

TOPIC 20

Programming

20.1 Overview

Numerous **videos** and **interactivities** are embedded just where you need them, at the point of learning, in your learnON title at www.jacplus.com.au. They will help you to learn the concepts covered in this topic.

20.1.1 Why learn this?

Computer programming is the process of creating instructions for a computer to follow to perform tasks. Many different languages can be used to program computers. These languages have specific purposes. For example, JavaScript is used in web pages to create interactive experiences like games; C is a computer language often embedded in programs in electronic devices like TVs and digital cameras.

This topic applies the computer language JavaScript in most of the examples and questions. JavaScript was chosen for its popularity and availability on nearly all web browsers.

According to Steve Jobs, the co-founder of Apple and Pixar, “everybody in this country [the USA] should learn how to program a computer because it teaches you how to think”.

20.1.2 What do you know?

assessment

- THINK** Write what you know about computer programming languages. Use a thinking tool such as a concept map to display your ideas.
- PAIR** Share what you know with a partner and then with a small group.
- SHARE** As a class, combine all the ideas into one large concept map or poster.

LEARNING SEQUENCE

- 20.1 Overview
- 20.2 Variables
- 20.3 Expressions
- 20.4 Sequences
- 20.5 Algorithms
- 20.6 Review



20.2 Variables

20.2.1 Values

- Variables in computer programs are similar to variables used in algebra. Variables are used to store values.
- In algebra, the value of a variable is fixed once it's defined. For example, the two equations $x = 2$ and $x = 3$ cannot be simultaneously true. Once x has been defined to be 2 ($x = 2$) it cannot be changed to be 3 ($x \neq 3$).
- In a computer program, the value of a variable can be changed at any point in the sequence. For example, the variable x is first assigned the value 2, and then changed to the value 3.
- Three of the most common types of values used in computer programming are:
 - **number** (a numerical value): 1, 2, -3, 1.5, 4.15359
 - **string** (any text in quotation marks "..."): "Some text", "123", "A string"
 - **Boolean** (used for logic; this type only has two possible values): true, false

WORKED EXAMPLE 1

Classify the following values as number, string or Boolean.

a "Aidan"

b 1.2

c "This is text"

d true

e "23"

f "true"

g ""

h -101

THINK

a "Aidan" is in quotation marks, so this value is a string.

b 1.2 is a numerical value not in quotation marks, so this value is a number.

c "This is text" is in quotation marks, so this value is a string.

d true is a Boolean value.

e "23" is in quotation marks, so it is a string.

f "true" is in quotation marks, so it is a string.

g "" is an empty pair of quotation marks, so it is a string.

h -101 is a numerical value, so it is a number.

WRITE

a String

b Number

c String

d Boolean

e String

f String

g String

h Number

20.2.2 Variable names

- A powerful feature of computers is their ability to store vast amounts of information accurately for any required amount of time. Computer languages use memory locations to store values. These named storage containers are called **variables**.
- Variables can have long or short names. Often longer names are used to remind the programmer the meaning of the information to be stored. The following are all valid variable names:
`x, y, i, age, name, first_name, number_of_people, bankBalance`
- There are complex rules that determine what can be used for a valid variable name. For simplicity, this topic restricts the variable names to three simple rules.
- Variable names:
 1. must not start with a number
 2. can only contain upper-case and lower-case letters, numbers and the underscore character (`_`), and cannot contain spaces
 3. cannot be JavaScript keywords.

Note: The following are JavaScript keywords, which should not be used as variable names.

abstract, arguments, Array, boolean, break, byte, case, catch, char, class, const, continue, Date, debugger, default, delete, do, double, else, enum, eval, export, extends, final, finally, float, for, function, goto, hasOwnProperty, if, implements, import, in, Infinity, instanceof, int, interface, isFinite, isNaN, isPrototypeOf, length, let, long, Math, name, NaN, native, new, null, Number, Object, package, private, protected, prototype, public, return, short, static, String, super, switch, synchronized, this, throw, throws, toString, transient, true, try, typeof, undefined, valueOf, var, void, volatile, while, with, yield, false

WORKED EXAMPLE 2

Is each of the following a valid variable name? (Answer either true or false for each name.)

- | | | |
|----------------|-----------|---------------|
| a a_variable | b 1_first | c %percentage |
| d camelCase_12 | e this | f function |

THINK

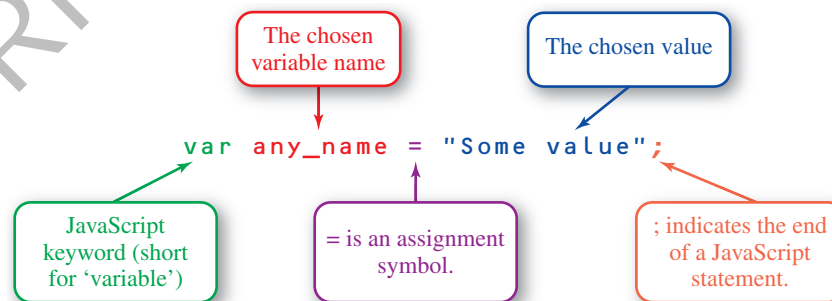
- a a_variable is valid because it only uses letters and the underscore character.
- b 1_first is invalid because it uses a number as the first character.
- c a%percentage is invalid because it uses a % character.
- d camelCase_12 is valid as it only uses letters, numbers and the underscore character. It also does not start with a number.
- e this is invalid because this is a JavaScript keyword.
- f function is invalid because function is a JavaScript keyword.

WRITE

- a True
- b False
- c False
- d True
- e False
- f False

20.2.3 Assigning values to variables

- Values are assigned to variables differently depending on the computer language used. When using JavaScript, the first time a value is assigned to a variable the instruction should begin with the JavaScript keyword `var`. JavaScript uses the following structure to assign a value to a variable.



WORKED EXAMPLE 3

Write a JavaScript statement to make the following variable assignments.

- Assign the value 10 to the variable `ten`.
- Assign `true` to the Boolean variable `I_can_write_code`.
- Assign `"JavaScript"` to the variable `language`.

THINK

a 1 Identify the variable name, which in this case is `ten`.

2 Identify the value, which in this case is `10`.

3 Apply the JavaScript structure to write the statement.

WRITE**a**

```
var ten = 10;
```

b 1 Identify the variable name, which in this case is `I_can_write_code`.

2 Identify the value, which in this case is `true`.

3 Apply the JavaScript structure to write the statement.

b

```
var I_can_write_code = true;
```

c 1 Identify the variable name, which in this case is `language`.

2 Identify the value, which in this case is `"JavaScript"`.


3 Apply the JavaScript structure to write the statement.

c

```
var language = "JavaScript";
```

learnon RESOURCES — ONLINE ONLY

 **Try out this interactivity:** CodeBlocks Activity 1
Searchlight ID: int-6573

 **Complete this digital doc:** How to write your own program
Searchlight ID: doc-18769

Exercise 20.2 Variables

assessment

Individual pathways

■ PRACTISE

Questions:
1–9, 11, 13

■ CONSOLIDATE

Questions:
1a–j, 2a–g, 3a–h, 4a–g, 5a–m,
6a–f, 8–13, 15

■ MASTER

Questions:
1k–q, 2h–m, 3i–o, 4h–k, 5n–w,
6g–j, 9, 10, 11d–g, 12g–k, 13–16

To answer questions online and to receive **immediate feedback** and **sample responses** for every question, go to your learnON title at www.jacplus.com.au. *Note:* Question numbers may vary slightly.

Fluency

1.  Classify the following values as number, string or Boolean.

a. 123

c. 3.6

e. "Is this a string?"

g. true

i. false

k. 917

m. "false"

o. ""

q. "Yes"

b. -12

d. "A string"

f. "10"

h. "animals"

j. "-10"

l. -10 000 000

n. "9999"

p. 0



2. Is each of the following a number value? (Answer either true or false for each value.)
- | | | |
|------------|----------------|------------------|
| a. "12" | b. "-13.22342" | c. 3.14159265359 |
| d. 1237123 | e. "1237123" | f. true |
| g. false | h. Three | i. "0" |
| j. Nine | k. Number12 | l. -80 |
| m. 4 | n. 0.0000001 | |

3. Is each of the following a string value? (Answer either true or false.)

- | | |
|---------------------------|-------------------------|
| a. "A string value" | b. Another string value |
| c. 12 | d. "12" |
| e. Four | f. "Four" |
| g. "true" | h. true |
| i. "false" | j. This is a string |
| k. "This is not a string" | l. false |
| m. Five | n. "Five" |
| o. 18276 | |

4. Is each of the following a Boolean value? (Answer either true or false.)

- | | | |
|----------|----------|-----------|
| a. true | b. false | c. "true" |
| d. 12 | e. Yes | f. No |
| g. "Yes" | h. "No" | i. 21344 |



5. **WE2** Variable names:

- must not start with a number
- can only contain upper-case and lower-case letters, numbers and the underscore character (`_`), and cannot contain spaces
- cannot be JavaScript keywords. The following are some JavaScript keywords, which should not be used as variable names: `true`, `var`, `false`.

Given these rules, is each of the following a valid variable name? (Answer either true or false.)

- | | |
|--|---------------------------------|
| a. variable | b. value |
| c. x1 | d. x2 |
| e. 1x | f. 2x |
| g. ten | h. 10 |
| i. percentage%variable | j. percentage_variable |
| k. fourteen | l. 14variable |
| m. ThisIsAVariableName | n. this_is_also_a_variable_name |
| o. 100_variable_names_sitting_in_a_row | |
| p. Variable name | q. a |
| r. i | s. j |
| t. _ | u. _private |
| v. true | w. false |

Understanding

6. Fill in the variable, value and type for the following JavaScript assignment statements.

	Variable	Value	Type	JavaScript statement
e.g.	<code>a_variable</code>	<code>"text"</code>	String	<code>var a_variable = "text";</code>
a				<code>var x = 10;</code>
b				<code>var nine = 9;</code>
c				<code>var aBoolean = true;</code>
d				<code>var _ = "A string";</code>

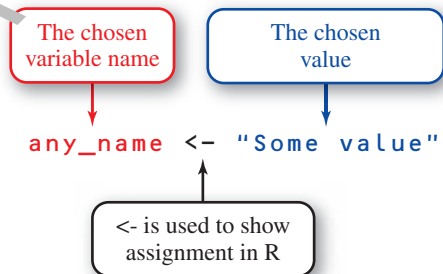
	Variable	Value	Type	JavaScript statement
e				<code>var _long_var = true;</code>
f				<code>var areYouTall = false;</code>
g				<code>var about_e = 2.718281828459;</code>
h				<code>var sin_90 = 1;</code>
i				<code>var isThis = "false";</code>
j				<code>var a_Boolean = true;</code>

- What are the two possible JavaScript Boolean values?
- Can a value assigned to a variable in a computer program be changed?
- WE3** Write a JavaScript statement to make the following variable assignments.
 - The value 10 assigned to the variable named `count`.
 - The value "Data" assigned to the variable called `information`.
 - The value 9 assigned to the variable `threeSquared`.
 - `true` assigned to the variable `isThisJavaScript`.
 - The variable `meaningOfLife` storing the value 42.
 - `classSize` holding the value 25.



- The computer language R has a different structure to assign values to variables, which is shown to the right. Rewrite the following JavaScript assignment statements in the computer language R.

- `var timeInSeconds = 120;`
- `var distanceInMiles = 40;`
- `var firstName = "Ben";`
- `var middleName = "Violet";`
- `var numberLeft = 19;`
- `var littleOverThree = 3.1;`



- Is each of the following a valid JavaScript assignment statement? (Answer either true or false for each.)
 - `var someVariable = 10;`
 - `var thisIsBoolean = "A string";`
 - `var time <- 10;`
 - `time <- 100`
 - `var aString = 10;`
 - `var thisLooksValid == 10;`
 - `var isThisBoolean = false;`

- The Python computer language uses slightly different Boolean values: `True` and `False`. Note that JavaScript's Boolean values are `true` and `false` (the case of the first letter is different). Python uses the assignment structure:

`variableName = "some value"`

Rewrite the JavaScript assignment statements shown in the table below as Python assignment statements.

	JavaScript assignment statement	Python assignment statement
a	<code>var x = 10;</code>	
b	<code>var nine = 9;</code>	

	JavaScript assignment statement	Python assignment statement
c	<code>var aBoolean = true;</code>	
d	<code>var _ = "A string";</code>	
e	<code>var _longer_variable = true;</code>	
f	<code>var areYouTall = false;</code>	
g	<code>var nearly4 = 3.99;</code>	
h	<code>var about_e = 2.718281828459;</code>	
i	<code>var sin_90 = 1;</code>	
j	<code>var isThisABoolean = "false";</code>	
k	<code>var is_this_a_Boolean = true;</code>	

Reasoning

13. Why is it important to have rules for the structure of variable names?

14. Variable names:

1. must not start with a number
2. can only contain upper-case and lower-case letters, numbers and the character `_`
3. cannot be JavaScript keywords like `var`, `true` and `false`.
 - i. Which of the rules listed do the following variable names violate?
 - ii. In each case explain why the rule is important

a. <code>true</code>	b. <code>false</code>	c. <code>1w</code>	d. <code>2</code>
e. <code>-value</code>	f. <code>variable name</code>	g. <code>"variableName"</code>	

Problem solving

15. Using only the characters `"a"`, `"b"`, `"_"` and `"1"`, list all the valid variable names:

- a. of length 1
- b. of length 2 without repeating characters
- c. of length 2, where characters can be repeated.

16. Variable names:

1. must not start with a number
2. can only contain upper and lower case letters, numbers and the underscore character (`_`), and cannot contain spaces
3. cannot be JavaScript keywords. The following are JavaScript keywords which should not be used as variable names.
`do, if, in, for, int, let, NaN, new, try, var, byte, case, char, Date, else, enum, eval, goto, long, Math, name, null, this, true, void, with, false`

Given these rules, how many possible valid variables names are there:

- | | | | |
|-----------------|-----------------|-----------------|-----------------|
| a. of length 1? | b. of length 2? | c. of length 3? | d. of length 4? |
|-----------------|-----------------|-----------------|-----------------|



Reflection

What is the difference between a number value, a string value and a Boolean value?

20.3 Expressions

20.3.1 Numerical expressions

- Expressions are used to perform calculations on existing values to generate new values.
- Computer programs use similar symbols and rules as standard mathematics to create numerical expressions.
- Numerical expressions involving numbers, brackets, plus and minus in JavaScript evaluate as you would expect in any mathematical expression.

WORKED EXAMPLE 4

Evaluate the JavaScript for the following expressions.

a $10 + 2$

b $5 - 3$

c $1.2 + 3.4$

d $3 + 6 - (5 - 4)$

THINK

a Add 10 and 2.

b Subtract 3 from 5.

c Add 1.2 and 3.4.

d Write out the expression.

Calculate the sub-expression $(5 - 4)$.

Calculate the final expression.

WRITE

a $10 + 2 = 12$

b $5 - 3 = 2$

c $1.2 + 3.4 = 4.6$

d $3 + 6 - (5 - 4) = 3 + 6 - (1)$
 $= 8$

- In JavaScript, for multiplication, use the character $*$; for division, use the character $/$.
- Fractions can be evaluated using the division character, but the numerator and denominator expressions need to be put into brackets.

$$\frac{2 + 3 + 1}{1 + 1} = (2 + 3 + 1) / (1 + 1)$$

Note: The characters $*$ and $/$ are only acceptable for use in programs and are not acceptable conventions for writing mathematics generally.

WORKED EXAMPLE 5

Write the following mathematical expressions as JavaScript expressions.

a 8×6

b $2 \times (5 - 3)$

c $12 \div (2 \times 2)$

d $\frac{12 + 3}{6 - 1}$

THINK

a Replace the \times with a $*$.

b Replace the \times with a $*$.

c Replace the \times with a $*$ and the \div with a $/$.

d The fraction is a division. The numerator and denominator require brackets.

WRITE

a $8 * 6$

b $2 * (5 - 3)$

c $12 / (2 * 2)$

d $(12 + 3) / (6 - 1)$

20.3.2 String expressions

- Expressions are not restricted to numerical expressions. Strings can also belong to expressions.
- Two or more strings can be concatenated (joined together) using the + symbol. For example, "Hello" + " " + "World" evaluates to "Hello World".

WORKED EXAMPLE 6

- Write a JavaScript expression to concatenate the strings "Strings", " ", "can", " have express", "ions".
- Write the expression.
- Write the string "I can write expressions" as an expression of single words and spaces.

THINK

- Replace the commas with +.
- Write the expression.
- Replace the spaces with " + " " + ".

WRITE

- "Strings" + " " + "can" + " have express" + "ions"
- "Strings can have expressions"
- "I" + " " + "can" + " " + "write" + " " + "expressions"

- A **character** is a string of length 1. A string is a list of characters. Each character position has an incrementing index starting at 0.

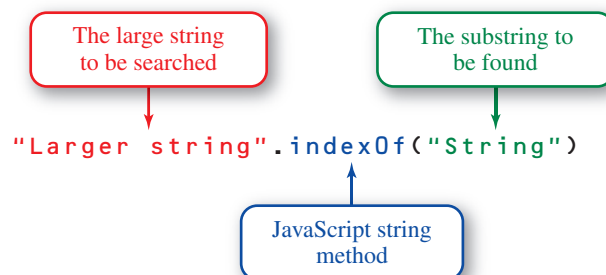
The string "a fox"	
Character	Index
"a"	0
" "	1
"f"	2
"o"	3
"x"	4

20.3.3 String methods

- JavaScript has several **methods** reserved for string expressions. These methods are used to manipulate a string.

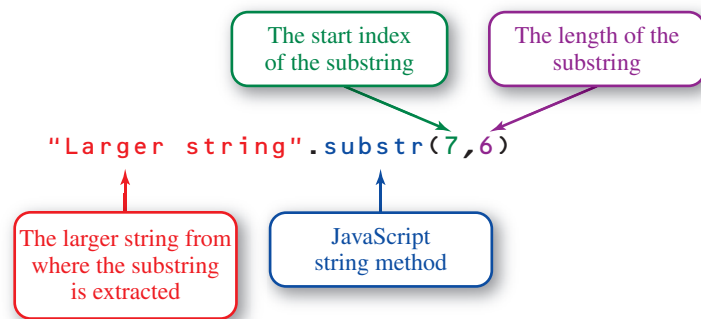
String indexOf method

- The JavaScript method `indexOf` is used to find the first occurrence of a substring inside a larger string. The value returned is an index (starting at 0) of the first occurrence. The `indexOf` method uses the structure shown in the following figure to define an expression.



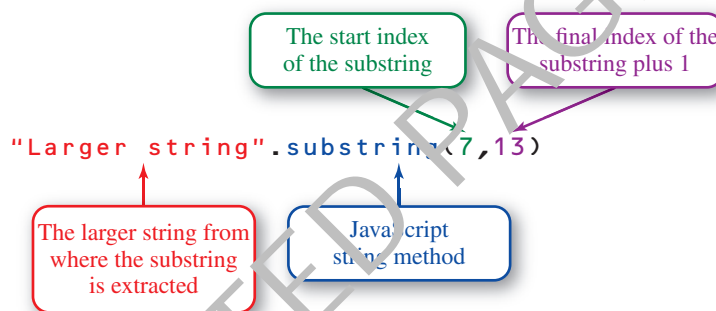
String substr method

- The JavaScript method `substr` extracts a substring from a larger string. The method requires the substring's starting index and length. The `substr` method uses the following structure to define an expression.



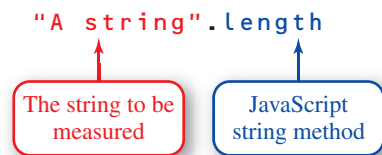
String substring method

- The JavaScript method `substring` also extracts a substring from a larger string. The method also requires the substring's starting index, but is different from the `substr` method because it requires the final index plus 1. The `substring` method uses the following structure to define an expression.



String length method

- The JavaScript method `length` measures the length of a string. The `length` method uses the following structure to define an expression.



WORKED EXAMPLE 7

Evaluate the following JavaScript expressions manually.

- `"Some text".indexOf("ex")`
- `"Some other".substr(6,3)`
- `"Some more".substring(6,8)`
- `"Final text".length`

THINK

- a Think about the index of each character. Remember that indexing starts from zero.

S	o	m	e		t	e	x	t
0	1	2	3	4	5	6	7	8

In "Some text" the first occurrence of "ex" starts at index 6.

S	o	m	e		t	e	x	t
0	1	2	3	4	5	6	7	8

- b Think about the index of each character.

S	o	m	e		o	t	h	e	r
0	1	2	3	4	5	6	7	8	9

Look at the first three characters starting at index 6.

S	o	m	e		o	t	h	e	r
0	1	2	3	4	5	6	7	8	9

- c Think about the index of each character.

S	o	m	e		m	o	r	e
0	1	2	3	4	5	6	7	8

Look at the characters starting at index 6 and ending at index $8 - 1 = 7$.

S	o	m	e		m	o	r	e
0	1	2	3	4	5	6	7	8

- d The number of characters in the string "Final text" is 10.

	F	i	n	a	l		t	e	x	t
Index	0	1	2	3	4	5	6	7	8	9
Length	1	2	3	4	5	6	7	8	9	10

WRITE

a

6

b

"the"

c

"or"

d 10

20.3.4 Assigning expressions

- Results of expressions can be stored as a variable.

WORKED EXAMPLE 8

Write a JavaScript statement to assign the following JavaScript expressions to the variable `store`.

a `27 / (12 - 3)`b `(6 + 4) * (1 + 12)`c `"Store lat" + "er"`**THINK**

Think about the assigning structure
`var store = "value";`
 required for JavaScript.

WRITE

a Replace "value" with the expression $27 / (12 - 3)$.

```
a var store = 27 / (12 - 3);
```

b Replace "value" with the expression $(6 + 4) * (1 + 12)$.

```
b var store = (6 + 4) * (1 + 12);
```

c Replace "value" with the expression "Store lat" + "er".

```
c var store = "Store lat" + "er";
```

learnon RESOURCES — ONLINE ONLY

Try out this interactivity: CodeBlocks Activity 2
Searchlight ID: int-6574

Exercise 20.3 Expressions

assessment

Individual pathways

PRACTISE

Questions:
1–5, 6a–h, 7, 8, 11

CONSOLIDATE

Questions:
1a–e, 2a–e, 3a–e, 4a–g, 5, 6a–h, 7,
8, 10a–c, 11a–c, 12, 14

MASTER

Questions:
1f–h, 2f–j, 3f–h, 4h–m, 5a–c, 6i–l,
7d–f, 8, 9, 10d–f, 11c–e, 12–15

To answer questions online and to receive **immediate feedback** and **sample responses** for every question, go to your learnON title at www.jacplus.com.au. *Note:* Question numbers may vary slightly.

Fluency

1. **WE4** Evaluate the following JavaScript expressions manually.

- $2 + 13$
- $52 - 3$
- $-21 + 34$
- $-12 - 13$
- $12 + (3 - 2)$
- $57 - (6 + 3)$
- $57 - (6 - 3)$
- $43 - 16 - (12 - (13 - 6)) + 2$

2. Evaluate the following JavaScript expressions manually.

- $20 / 5$
- $5 * 3$
- $2 * 2$
- $(2 + 13) / 5$
- $(7 + 12) / (2 + 3)$
- $100 / 25 - 9 / 3$
- $5 * 3 + 16 / 4$
- $8 + 6 * 2 / 3$
- $((12 - 3) / (5 - 2) + 16 / 8 - 4) / 2$
- $(-48 / 8 - 4) / 2 + (22 - 13) / (15 / 5)$

3. Evaluate the following JavaScript expressions manually.

- "Quotation marks" + " at " + "the start" + " and end"



- b. "This " + "is a" + " string"
 c. "Hello " + "World" + "!"
 d. "Express" + "ions ma" + "ke strings f" + "un"
 e. "Happy :" + ")"
 f. "Java" + "Script"
 g. ("String " + "expressions") + (" using" + " brackets")
 h. "Remember" + " the two " + " spaces"
4. **WE7** Evaluate the following JavaScript expressions manually.
- a. "Four".length
 b. ("Java" + "Script").length
 c. "Java".length + "Script".length
 d. "The quick brown fox jumps over the lazy dog".length
 e. "Moving forward".indexOf("i")
 f. "The quick brown fox".length
 g. "Find the e".indexOf("e")
 h. "Own this gown".indexOf("own")
 i. "The quick brown fox".substr(4,5)
 j. "The quick brown fox".substr(12,3)
 k. "The quick brown fox".substring(4,5)
 l. ("The sub" + "string is the answer").substring(4,13)
 m. "The quick brown fox".substr(4,8).length
5. Evaluate the following JavaScript expressions manually.
- a. "Find the four".substr(4 + 1 + 3 + 1,4)
 b. "Find five wait find four".substr(4 * 5, "four".length)
 c. "Take apart put together".substr(0,22 / 2) + "put together"
 d. "Stop here".indexOf("e")
 e. ("The sub" + "string is the answer").substring(4,39 / 3)

Understanding

6. **WE5** Write the following mathematical expressions as JavaScript expressions

a. 6×11

c. $\frac{12}{5}$

e. $(2 - 4)(-3 + 5)$

g. $3 \times 7 + 3$

i. $\frac{12 + 2(4 - 2)}{5}$

k. $\frac{(2 + 4)(-2 + 6)}{3}$

b. $54 \div 3$

d. $6(2 + 3)$

f. $\frac{7 + 2}{4 - 1}$

h. $100 \div 25 + 4 + 4 \times 3$

j. $6(2 + 5)(4 - 1) - 3$

l. $\frac{2 + 2}{3 - 2} \times \frac{1 + 2}{4 - 1}$

7. Write the expression to extract the string "and" from the string below with the method:

i. `substr`

ii. `substring`

a. "This fox and sheep"

c. "Use hands"

e. "ksanbjhandasdkjh"

b. "Find and back there"

d. "Search the land"

f. "Earth and Mars"



8. `charAt(index)` is the same as `substr(index, 1)`.
`charAt` returns one character at the specified index. For example,
`"Find the letter h".charAt(6)` evaluates to `"h"`.
 Evaluate the following JavaScript expressions.
- `"The letter x is at index 23".charAt(23)`
 - `"Where is z hiding?".charAt(9)`
 - `"Pick 5 today".charAt(5)`
 - `"Find the last character.".charAt("Find the last character.".length-1)`
9. There are many more string methods that can be used in expressions.
`toUpperCase` converts all lower-case letters to upper-case letters.
`toLowerCase` converts all the upper-case letters to lower-case letters.
`"To upper case".toUpperCase()` evaluates to `"TO UPPER CASE"`.
`"To Lower Case".toLowerCase()` evaluates to `"to lower case"`.
 Evaluate the following JavaScript expressions manually.
- `"Make all upper case".toUpperCase()`
 - `"Make ALL lower case".toLowerCase()`
 - `"A title".toUpperCase()`
 - `"STOP SHOUTING".toLowerCase()`
10. The `%` symbol is used to find the remainder after an integer division. `7%3` evaluates to 1, as 3 divides into 7 twice with a 1 remainder.
 Evaluate the following JavaScript expressions manually.
- `20%3`
 - `19%5`
 - `200%8`
 - `12%5`
 - `6%5`
 - `3%120`
11. **WEB** Write a JavaScript statement to assign the following JavaScript expressions to the variable `finalResult`.
- `27 - (16 - 3)`
 - `(3 + 23) / (10 + 3)`
 - `"Assign" + "ed to finalResult"`
 - `"Len".length + "gth".length`
 - `18%5`

Reasoning

12. Write a JavaScript expression and assign to the stated variable for the following situations.
- Tom has 12 toy cars and 2 friends. Write a numerical expression to divide the cars between the three of them and assign it to the variable `numberOfCarsEach`.
 - Kate has saved \$20 and earned \$15 today. Write an expression to evaluate how much money she has now in dollars and assign it to the variable `totalCash`.
 - Vanessa wrote the text "My Trip" and she would like to change all the letters to upper case using the string method `toUpperCase`. Write an expression to convert the letters to upper case and assign it to the variable `myTitle`.
13. Write a JavaScript expression and assign it to the variable `questionResult` for the following questions.
- Calculate $10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$.



- b. Add the even numbers between 1 to 19.
- c. Join the strings "Th", "e", " f", "in", "al re", "sult is 2", "2" and then calculate the length.

Problem solving

14. Rearrange all the characters 123456789* to form a valid expression:
 - a. that evaluates to the smallest possible number.
 - b. that evaluates to the largest possible number.
15. Given an English word, write a JavaScript program to translate the word into Pig Latin. Output the translated Pig Latin to the console. To translate a word into Pig Latin, move the first letter to the end of the word and then add "ay". For example, "javascript" becomes "avascriptjay". The program has been started below.


```
var englishWord = "program";
```

Reflection

How will you remember the different string methods?

20.4 Sequences

20.4.1 Comments

- Computer programs are usually made up of more than one statement. This list of statements is run in order as a sequence.
- Knowing a computer language is often not enough for people to understand the code of a program easily.
- Comments are added to the code to give hints about the function of the program. The comments are completely ignored by the computer, so are only for the benefit of the reader.
- JavaScript comments start with `//` and the rest of the text on that line is ignored by the computer. For example

```
var x = 2; // The smallest prime number.
```

- The first part of the line `var x = 2;` is the actual JavaScript code to be run.
- The second part `// The smallest prime number` is ignored by the computer and is only there for the reader's reference. The text could be changed and it would make no difference to the running of the program.

WORKED EXAMPLE 9

Add a comment to each statement of the form `// variable = value` where `variable` is the variable name and `value` is the evaluated value.

```
a var a = 2 + 1 / 10;          b var time = 30 + 15;
c var F5 = 1 * 2 * 3 * 4 * 5;
```

THINK

- 1 The variable name is a.
- 2 The calculated value is $2 + 1/10 = 2.1$.

WRITE

a

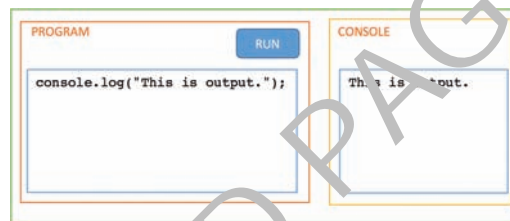
- 3 Add the comment to the statement. `var a = 2 + 1 / 10; // a = 2.1`

- b** 1 The variable name is `time`. **b**
- 2 The calculated value is
 $30 + 15 = 45$.
- 3 Add the comment to the statement. `var time = 30 + 15; // time = 45`

- c** 1 The variable name is `F5`. **c**
- 2 The calculated value is
 $1 \times 2 \times 3 \times 4 \times 5 = 120$.
- 3 Add the comment to the statement. `var F5 = 1 * 2 * 3 * 4 * 5; // F5 = 120`

20.4.2 Output

- The **console** is a special region in a web browser for monitoring the running of JavaScript programs. Most web browsers, including Chrome, Firefox, Safari, Internet Explorer, Microsoft Edge and Opera, allow you activate the console through the menu options.
- The JavaScript `console.log` function can be used to see the result of an expression or value. This function outputs results to the console. A diagram of this using the given statement and output is shown on the next page.



- Statement: `console.log("This is output in the console");`
 - Output: `This is the output in the console`
- Note:* The quotation marks in strings are not shown in the output.

WORKED EXAMPLE 10

What is the output to the console after running the following statements?

- a** `console.log((16 + 4)/(11 - 1));`
b `console.log(("Test" + " " + "this").length);`
c `console.log("Con" + "sole out" + "put");`

THINK

- a** 1 The expression $(16 + 4) / (11 - 1)$ equals 2.
 2 The statement simplifies to `console.log(2)`.
- b** 1 The expression `"Test" + " " + "this"` equals `"Test this"`.
`"Test this"` has a length of 9.
 2 The statement simplifies to `console.log(9)`.
- c** 1 The expression `"Con" + "sole out" + "put"` equals `"Console output"`.
 2 The statement simplifies to `console.log("Console output")`.
 3 The `console.log` output does not show the quotation marks.

WRITE

a

2

b

9

c

**Console
output**

WORKED EXAMPLE 11

- i Write a JavaScript statement to output to the console the following expressions.
- ii Write the output.

a $(2 + 12) / (9 - 2)$

b `"putter".substr(0,3)`

THINK

Think about the structure of the `console.log()` statement.

WRITE

a i Put the expression inside the brackets.

a i `console.log((2 + 12) / (9 - 2));`

ii Write the output.

ii 2

b i Put the expression inside the brackets.

b i `console.log("putter".substr(0,3));`

ii Write the output.

ii put

20.4.3 Store and use a result

- A common sequence of statements is to:
 1. calculate an expression and store the result in a variable
 2. use the stored result in a new expression.

```
var storedForLater = "An expression";  
var newString = storedForLater + "with more";  
// Output "An expression with more" console.log(newString);
```

WORKED EXAMPLE 12

For each statement in the following programs:

- find the variable name between `var` and `=`
- calculate the value assigned to this variable.

Append a comment to each statement indicating the progress. The comment should be of the form `// variable = value`.

a `var r = 10 / 2;`

`var area = r * r;`

b `var a = 2 * 5;`

`var b = 20 + a;`

`var c = 30 + b;`

`var d = 4 * a * c - b * b;`

c `var y = 2 + 8;`

`var y = 3 + 4;`

`var x = 3 * y;`

THINK

- a 1** The variable is `r`. $10 / 2$ is 5.
2 The variable is `area`. $5 * 5$ is 25.
- b 1** The variable is `a`. $2 * 5$ is 10.
2 The variable is `b`. $20 + 10$ is 30.
3 The variable is `c`. $30 + 30$ is 60.
4 The variable is `d`.
 $4 * 10 * 60 - 30 * 30$ is 1500.
- c 1** The variable is `y`. $2 + 8$ is 10.
2 The variable is `y`. $3 + 4$ is 7.
`y` now has a new value and the old value is lost from memory.
3 The variable is `x`. $3 * 7$ is 21.

WRITE

```
a var r = 10 / 2; // r = 5
   var area = r * r; // area = 25
b var a = 2 * 5; // a = 10
   var b = 20 + a; // b = 30
   var c = 30 + b; // c = 60
   var d = 4 * a * c - b * b; // d = 1500
c var y = 2 + 8; // y = 10
   var y = 3 + 4; // y = 7
   var x = 3 * y; // x = 21
```

WORKED EXAMPLE 13

Manually simulate running the programs shown below. What is the final value stored in the variable `x`?

a `var a = 10 + 2;`

`var x = a * a;`

b `var x1 = "one";`

`var x2 = "two";`

`var s = " ";`

`var x = x1 + s + x2;`

c `var r = 4;`

`var r = 6;`

`var x = r + r;`

THINK

- a 1** Write the equation.
2 Perform the addition.
3 Write the equation (use `*` not `x`).
4 Substitute 12 for `a`.
5 Perform the multiplication.
- b 1** Write the first assignment.
2 Write the second assignment.
3 Write the third assignment.
4 Write the fourth equation.
5 Substitute values into `x1`, `x2` and `s`.
6 Evaluate the expression.

WRITE

a `a = 10 + 2`

`a = 12`

`x = a * a`

`x = 12 * 12`

`x = 144`

b `x1 = "one"`

`x2 = "two"`

`s = " "`

`x = x1 + s + x2`

`x = "one" + " " + "two"`

`x = "one two"`

- c 1 Write the assignment.
- 2 Write the reassignment.
- 3 Write the equation.
- 4 Substitute 6 for r.
- 5 Evaluate the expression.

$$\begin{aligned}c r &= 4 \\ r &= 6 \\ x &= r + r \\ x &= 6 + 6 \\ x &= 12\end{aligned}$$

20.4.4 Sequence

- The sequence of statements in a program is very important. Rearranging statements in a program can change the end result and/or cause an error.

WORKED EXAMPLE 14

- a Find all the different programs that can be created from rearranging the three statements in the following table.

Reference	Statement
A	<code>var a = 2;</code>
B	<code>a = a + 3;</code>
C	<code>a = a * a;</code>

- b Add `console.log(a);` to the end of each program.
- c Find the end result for each of the programs.

THINK

- a There six possible ways to arrange three statements.

ABC	ACB
BAC	BCA
CAB	CBA

- b Add the `console.log(a)` statement to the end of each program.

WRITE

A	<code>var a = 2;</code>	A	<code>var a = 2;</code>
B	<code>a = a + 3;</code>	C	<code>a = a * a;</code>
C	<code>a = a * a;</code>	B	<code>a = a + 3;</code>
B	<code>a = a + 3;</code>	B	<code>a = a + 3;</code>
A	<code>var a = 2;</code>	C	<code>a = a * a;</code>
C	<code>a = a * a;</code>	A	<code>var a = 2;</code>
C	<code>a = a * a;</code>	C	<code>a = a * a;</code>
A	<code>var a = 2;</code>	B	<code>a = a + 3;</code>
B	<code>a = a + 3;</code>	A	<code>var a = 2;</code>

A	<code>var a = 2;</code>	A	<code>var a = 2;</code>
B	<code>a = a + 3;</code>	C	<code>a = a * a;</code>
C	<code>a = a * a;</code>	B	<code>a = a + 3;</code>
	<code>console.log(a);</code>		<code>console.log(a);</code>
B	<code>a = a + 3;</code>	B	<code>a = a + 3;</code>
A	<code>var a = 2;</code>	C	<code>a = a * a;</code>
C	<code>a = a * a;</code>	A	<code>var a = 2;</code>
	<code>console.log(a);</code>		<code>console.log(a);</code>
C	<code>a = a * a;</code>	C	<code>a = a * a;</code>
A	<code>var a = 2;</code>	B	<code>a = a + 3;</code>
B	<code>a = a + 3;</code>	A	<code>var a = 2;</code>
	<code>console.log(a);</code>		<code>console.log(a);</code>

c For each program, take the value of a through each step.

A	<code>var a = 2;</code>	A	<code>var a = 2;</code>
B	<code>a = 2 + 3;</code>	C	<code>a = 2 * 2;</code>
C	<code>a = 5 * 5;</code>	B	<code>a = 4 + 3;</code>
	The final result is 25.		The final result is 7.
B	<code>a = a + 3;</code>	B	<code>a = a + 3;</code>
A	<code>var a = 2;</code>	C	<code>a = a * a;</code>
C	<code>a = 2 * 2;</code>	A	<code>var a = 2;</code>
	The final result is 4.		The final result is 2.
C	<code>a = a * a;</code>	C	<code>a = a * a;</code>
A	<code>var a = 2;</code>	B	<code>a = a + 3;</code>
B	<code>a = 2 + 3;</code>	A	<code>var a = 2;</code>
	The final result is 5.		The final result is 2.

20.4.5 Computer languages

- Computers follow instructions called computer programs, which are written in a computer language.
- Each computer language has its own clear set of rules and structure called a syntax.
- There are many different computer languages. Some examples include JavaScript, TypeScript, Python, Ruby, C#, C, C++, R and PHP.
- Below are sections of programs using different computer languages. All these programs perform the same key tasks:
 - Assign the value 10 to the variable x.
 - Calculate $2 + x$ (x is currently storing the value 10).
 - Evaluate this to get the value 12.
 - Assign the evaluated value 12 to the variable y.

JavaScript <code>var x = 10;</code> <code>var y = 2 + x;</code>	TypeScript <code>var x:number = 10;</code> <code>var y:number = 2 + x;</code>	Python and Ruby <code>x = 10</code> <code>y = 2 + x</code>
C#, C and C++ <code>int x = 10;</code> <code>int y = 2 + x;</code>	R <code>x <- 10</code> <code>y <- 2 + x</code>	PHP <code>\$x = 10;</code> <code>\$y = 2 + \$x;</code>

WORKED EXAMPLE 15

Consider the following R program.

```
p <- 4
r <- p + 3
```

Rewrite the program in JavaScript.

THINK

Compare the two example programs.

WRITE

R	JavaScript
<code>x <- 10</code>	<code>var x = 10;</code>
<code>y <- 2 + x</code>	<code>var y = 2 + x;</code>

R	JavaScript
<code>p <- 4</code>	<code>var p = 4;</code>
<code>r <- p + 3</code>	<code>var y = p + 3;</code>

Note the changes required to go from R to JavaScript statements.

- Add "var" to the start.
- Change "<-" to "=".
- Append ";" to each statement.

learnON RESOURCES — ONLINE ONLY

Try out this interactivity: CodeBlocks Activity 3
Searchlight ID: int-6575

Exercise 20.4 Sequences

assessment

Individual pathways

PRACTISE

Questions:
1–8, 12, 14

CONSOLIDATE

Questions:
1, 2a–c, 3, 4a–d, 5, 6, 7a–c, 8, 9,
12, 14, 15

MASTER

Questions:
2d–f, 3, 4d–f, 6b, c, 7c–e, 8–16

To answer questions online and to receive **immediate feedback** and **sample responses** for every question, go to your learnON title at www.jacplus.com.au. *Note:* Question numbers may vary slightly.

Fluency

1. The program statements in the table below relate to a rectangle. Match the following JavaScript comments with the appropriate statements.

```
// set the height to 15  
// set the width to 10  
// calculate half the perimeter  
// calculate the area  
// calculate the perimeter
```

Program	Comment
<code>var width = 10;</code>	
<code>var height = 15;</code>	
<code>var halfAround = width + height;</code>	
<code>var allAround = 2 * halfAround;</code>	
<code>var area = width*height;</code>	

2. **NOTE** What is the output to the console after running the following statements?

- `console.log((18 + 2)/(6 - 1));`
- `console.log(("How long is" + " " + "this").length);`
- `console.log("No" +
"quotation marks in console output");`
- `console.log(("This is d").substring(8,9));`
- `console.log(("Measure the length").length);`
- `console.log(("The length is 16").substr(14,2));`

3. **WE13** Manually simulate running the following programs. What is the final value stored in the variable z?

```

a. var y = 14 + 16;
   var z = y / 3;
b. var x1 = "cat";
   var x2 = "jumped";
   var x3 = "slept";
   var s = " ";
   var z = x1 + s + x3;
c. var radius = 1;
   var radius = 22;
   var z = radius + radius;

```



4. Manually simulate running the following programs. What will be the output to the console?

```

a. var a = 10;
   console.log(a * a + 5);
b. var aString = "Ten";
   console.log(aString.length * 3 + 1);
c. var a = 3;
   var b = 4;
   var c = 5;
   console.log(a * a + b * b - c * c);
d. var single = "One";
   var double = "two";
   var three = single + " plus " + double;
   console.log(three);
e. var x = "10";
   var y = "20";
   console.log(x + y);
f. var x = 10;
   var y = 20;
   var z = x + y;
   console.log(y);

```

5. Match the JavaScript statements below with the equivalent C++ statements in the table.

```

var aVariable = 6 * 8 * 7 * 9;
var x = 10 + 3;
var y = 3 + 10;
var x = 10 + 4;
var x = 10 * 4;

```

C++ statements	JavaScript statements
int y = 10 + 3;	
int x = 10 + 4;	
int x = 3 + 10;	
int x = 10 * 4;	
int aVariable = 9 * 8 * 7 * 6;	

Understanding

6. **WE9** To each statement, add a comment of the form `// variable = value`, where `variable` is the variable name and `value` is the evaluated value.
- `var nearly_e = 2 + 7 / 10 + 1 / 100 + 8 / 1000;`
 - `var marathonKm = 26 * 1.6;`
 - `var secondsPerDay = 60 * 60 * 24;`
7. **WE11** Write a JavaScript statement to output the following expressions to the console.
- `(3 + 41) / (9 + 2)`
 - `"putter".substr(1,5)`
 - `2 + 3 - 5`
 - `(2 * 12)/(8 - 2)`
 - `"life".substring(1,3)`
8. **WE12** For each statement in the programs below:
- find the `variable` name between `var` and `=`
 - calculate the `value` assigned to this `variable`.
- Append a comment of the form `// variable=value` to each statement indicating the progress.
- `var side1 = 3 * 4;`
`var side2 = side1 / 2;`
`var area = side1 * side2;`
 - `var a = 5 + 2;`
`var b = a + a;`
`var c = b + b;`
`var delta = 4 * a * c - b * b;`
 - `var reassign = 12 + 18;`
`var reassign = 1 + 41;`
`var result = 3 * reassign;`
9. **WE15** Rewrite the following R program in JavaScript.
- ```
r <- 8
r_less_5 <- r - 5
```
10. a. Rewrite the following JavaScript program in C.
- ```
var minutes = 3;  
var seconds = minutes * 60;
```
- b. Write a JavaScript program to convert 3 hours into minutes.
11. What is the most likely programming language of the following statements?
- `var thisTyped:number = 101;`
 - `storeM.Here <- 19`
 - `$dollars = 10 + 13;`
 - `var webLanguage = "JavaSc" + "ript";`
 - `var newType:string = "Another type of " + "script";`
 - `console.log("Nothing new here");`
12. **WE14** a. Find all the different programs that can be created from rearranging the following three statements.

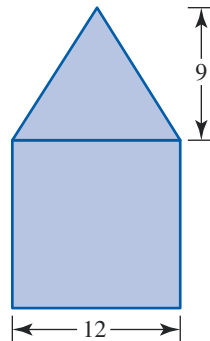
Reference	Statement
A	<code>var e = 5;</code>
B	<code>e = (e + 7) * 2;</code>
C	<code>e = e / e;</code>

- Add `console.log(e);` to the end of each program.
- Find the end result for each of the programs.



Reasoning

13. Write a JavaScript program to calculate the **perimeter** of a rectangle with a **width** of 30 units and **length** of 20 units.
14. Fill in the gaps for the following JavaScript program to calculate the area of a composite shape. The shape is made up of a square of



width 12 units and triangle of height 9 units.

```
var _____ = 12;  
var _____ = 9;  
var triangleArea = height * _____ / 2;  
var squareArea = _____ * width;  
var totalArea = squareArea + _____;
```

Problem solving

15. Rearrange the following program so that it starts with $x = 3$ and finishes with $x = 100$.

Hint: Assume x is always a positive whole number through the entire program sequence.

```
var x = 3;  
x = x / 24;  
x = x - 96;  
x = x + 9;  
x = 10 * x; (Hint: This step is in the correct position.)  
x = 110 - x;  
x = x - 10;  
x = x * x;  
x = 10
```

16. Fill in the single character gaps below using the nine digits 123456789 so the program finishes with $x = 1$. The 8 has already been filled in.

Hint: Assume x is always a positive whole number through the entire program sequence.

```
var x = _;  
x = x / _;  
x = x + _;  
x = x / _;  
x = x + 8;  
x = x / _;  
x = x + _;  
x = x - _;  
x = _ - x;  
x = 1
```

Reflection

Why is the sequence of steps so important in a program?

20.5 Algorithms

20.5.1 Defining a function

- In JavaScript a function is used to give a **name** to a section of code. The code below shows an example of the structure used to do this.

```
function name() {  
    statement 1  
    statement 2  
    statement n  
}
```

Note: All functions start and end with curly braces { } as shown in the example above.

WORKED EXAMPLE 16

- What is the name of each of the following functions?
- What is the output to the console if the code inside the function was run?

```
a function triangleArea() {  
    var base = 10;  
    var height = 3;  
    console.log(base * height / 2);  
}  
b function outputVolume() {  
    var width = 3;  
    var height = 4;  
    var depth = 5;  
    console.log(width * height * depth);  
}  
c function complexFormula() {  
    var w = 6;  
    var h = 7;  
    var d = 8;  
    console.log(2 * w * h + 2 * w * d + 2 * h * d);  
}
```

THINK

- Identify the **name** after the function keyword.
- Write the first assignment.
 - Write the second assignment.
 - Write the output expression.
 - Substitute 10 for **base** and 3 for **height**.
 - Evaluate the output expression.

WRITE

```
a i triangleArea  
ii base = 10  
    height = 3  
    base * height / 2  
    = 10 * 3 / 2  
    = 15
```

b i Identify the **name** after the function keyword.

ii 1 Write the first assignment.

2 Write the second assignment.

3 Write the third assignment.

4 Write the output expression.

5 Substitute **3** for **width**, **4** for **height** and **5** for **depth**.

6 Evaluate the output expression.

c i Identify the **name** after the function keyword.

ii 1 Write the first assignment.

2 Write the second assignment.

3 Write the third assignment.

4 Write the output expression.

5 Substitute **6** for **w**, **7** for **h** and **8** for **d**.

6 Evaluate the output expression.

b i `outputVolume`

ii `width = 3`

`height = 4`

`depth = 5`

`width * height * depth`

`= 3 * 4 * 5`

`= 60`

c i `complexFormula`

ii `w = 6`

`h = 7`

`d = 8`

`2 * w * h + 2 * w * d`

`+ 2 * h * d`

`= 2 * 6 * 7 + 2 * 6 * 8`

`+ 2 * 7 * 8`

`= 292`

20.5.2 Calling a function

- Defining a function only tells the computer what to do if the function is **called**. In order for a function to run, it needs to be called with a separate statement.
- Once a function with the **name** has been defined, it can be called with the statement `name()`.

WORKED EXAMPLE 17

Given the following function names, what is the statement to call the function?

a `outputTime`

b `updateResults`

c `createNew`

THINK

a Identify the function **name** and join with `()`.

b Identify the function **name** and join with `()`.

c Identify the function **name** and join with `()`.

WRITE

a `outputTime();`

b `updateResults();`

c `createNew();`

- To be a valid program, a function must be defined before it can be called. A function can be called many times once defined, as shown in the following example code.


```
function name() {
  statement 1
  statement 2
  statement n
}
name();
name();
name();
```

WORKED EXAMPLE 18

Are the following valid programs? (Answer either true or false.)

- a `function something() {
 console.log("Output");
}
something();
something();`
- b `function doesNothing() {
}
doesNothing();`
- c `wrongOrder();
function wrongOrder() {
 console.log("Does not work");
}`
- d `function function1() {
 console.log("Function 1");
}
console.log("In the middle");
function1();`

THINK

- a This program is valid as `something` is defined before being called.
- b This program is valid as `doesNothing` is defined before being called.
- c This program is invalid as `wrongOrder` is called before being defined.
- d This program is valid as `function1` is defined before being called. A program is still valid if it has a statement between the definition and calling statement.

WRITE

- a True
- b True
- c False
- d True

WORKED EXAMPLE 19

- a Write a function named `outputResults` to output "Some results" to the console.
- b Write a statement to call the function.
- c Combine the function definition and statement into one program.
- d Write the output.

THINK

a Identify the name of the function.

Write the output statement.

Finalise the function definition.

b Write the calling statement by adding `()` to the end of the function.

c Write the function definition.

Write the calling statement.

d Write the output.

WRITE

```
a function outputResults() {
    console.log("Some results");
}
```

```
b outputResults();
```

```
c function outputResults() {
    console.log("Some results");
}
```

```
outputResults();
```

```
d Some results
```

20.5.3 Input

- In JavaScript a function can be given an input to vary the function's behaviour as shown below.

```
function name(input) {
    statement 1
    statement 2
    statement n
}
```

WORKED EXAMPLE 20

i What is the name of each of the following functions?

ii What is the name of the input?

iii What is the output to the console if the functions were called with the input value 10?

```
a function outputDouble(inputValue) {
    console.log(2 * inputValue);
}
```

```
b function outputVolume(side) {
    console.log(side * side * side);
}
```

```
c function sixTimesSquared(x) {
    var squared = x * x;
    console.log(6 * squared);
}
```

THINK

a i Identify the name after the function keyword.

ii Identify the input inside the function brackets `()`.

WRITE

a i `outputDouble`

ii `inputValue`

iii 1 Write the output statement.	iii <code>console.log(2 * inputValue)</code>
2 Substitute 10 for <code>inputValue</code> .	<code>console.log(2 * 10)</code>
3 Evaluate the expression.	<code>console.log(20)</code>
4 Write the output to console.	20
b i Identify the <code>name</code> after the <code>function</code> keyword.	b i <code>outputVolume</code>
ii Identify the <code>input</code> inside the function brackets <code>()</code> .	ii <code>side</code>
iii 1 Write the output statement.	iii <code>console.log(side * side * side)</code>
2 Substitute 10 for <code>side</code> .	<code>console.log(10 * 10 * 10)</code>
3 Evaluate the expression.	<code>console.log(1000)</code>
4 Write the output to console.	1000
c i Identify the <code>name</code> after the <code>function</code> keyword.	c i <code>sixTimesSquared</code>
ii Identify the <code>input</code> inside the function brackets <code>()</code> .	ii <code>x</code>
iii 1 Write the assignment expression	iii <code>squared = x * x</code>
2 Substitute 10 for <code>x</code> .	<code>squared = 10 * 10</code>
3 Evaluate the assignment.	<code>squared = 100</code>
4 Write the output statement.	<code>console.log(6 * squared)</code>
5 Substitute 100 for <code>squared</code> .	<code>console.log(6 * 100)</code>
6 Evaluate the expression.	<code>console.log(600)</code>
7 Write the output to console.	600

- In JavaScript a function can be given zero, one or more inputs; for example, the inputs could be `x1, x2, ...` as shown in the following code. The inputs are given as a list, separated by commas.

```
function name(x1,x2,...) {
    statement 1
    statement 2
    statement n
}
```

WORKED EXAMPLE 21

- i** How many inputs do each of the following functions have?
ii What are the names of the inputs? (Write these in order, with no spaces and separated with commas.)

a

```
function outputSquared(ignore,x) {  
  console.log(x * x);  
}
```

b

```
function outputVolume(w,h,d) {  
  console.log(w * h * d);  
}
```

c

```
function requiresNothing() {  
  console.log("Always the same");  
}
```

THINK

- a** **i** Count the inputs inside the function brackets ().
ii Identify the inputs inside the function brackets ().
- b** **i** Count the inputs inside the function brackets ().
ii Identify the inputs inside the function brackets ().
- c** **i** Count the inputs inside the function brackets ().
ii Identify that there are no inputs inside the function brackets ().

WRITE

- a** **i** 2
ii ignore, x
- b** **i** 3
ii w, h, d
- c** **i** 0
ii There are no inputs.

- A function with inputs x_1, x_2, \dots can be called with the corresponding values v_1, v_2, \dots . The value v_1 is assigned to the input x_1 , the value v_2 is assigned to the input x_2 , etc.

```
function name(x1,x2,...) {  
  statement 1  
  statement 2  
  statement n  
}  
name(v1,v2,...);
```

WORKED EXAMPLE 22

What is the output to the console of each of the following programs?

a

```
function outputCubed(x) {  
  console.log(x * x * x);  
}  
outputCubed(3);
```

```

b function rectangleArea(w,h) {
    console.log(w * h);
}
rectangleArea(11,2);
c function noInputs() {
    console.log("Never changes");
}
noInputs();

```

THINK

- a**
- 1 The input `x` equals 3.
 - 2 Write the output statement.
 - 3 Substitute 3 for `x`.
 - 4 Evaluate the expression
 - 5 Write the output to console.
- b**
- 1 The input `w` equals 11.
 - 2 The input `h` equals 2.
 - 3 Write the output statement.
 - 4 Substitute 11 for `w` and 2 for `h`.
 - 5 Evaluate the expression.
 - 6 Write the output to console.
- c**
- 1 The function has no inputs.
 - 2 Write the output statement
 - 3 Write the output to console.

WRITE

```

a x = 3
  console.log(x * x * x)
  console.log(3 * 3 * 3)
  console.log(27)
  27
b w = 11
  h = 2
  console.log(w * h)
  console.log(11 * 2)
  console.log(22)
  22
c
  console.log("Never changes")
  Never changes

```

20.5.4 Return value

- A JavaScript function will execute until a return statement is reached, at which point the function stops running the section of code and output is returned to the 'caller'. The output can then be stored in a variable `stored`. The structure of the function and storing the result is shown below.

```

function name(x1,x2,...) {
    statement 1
    statement 2
    statement n
    return output;
}
var stored = name(v1,v2,...);

```

WORKED EXAMPLE 23

Manually simulate running each of the following programs. What is the final value stored in the variable x ?

```
a function volumePrism(height,area) {  
    return height * area;  
}
```

```
var x = volumePrism(10,40);
```

```
b function mean(x1,x2,x3,x4) {  
    total = x1 + x2 + x3 + x4;  
    return total / 4;  
}
```

```
var x = mean(2,4,6,8);
```

THINK

- a**
- 1 Write the assignment.
 - 2 The input h equals 10.
 - 3 The input $area$ equals 40.
 - 4 Write the return expression.
 - 5 Substitute 10 for h and 40 for $area$.
 - 6 Evaluate the return expression.
 - 7 Write the assignment.
 - 8 Substitute 400 for $volumePrism(10,40)$.
- b**
- 1 Write the assignment.
 - 2 The input x_1 equals 2.
 - 3 The input x_2 equals 4.
 - 4 The input x_3 equals 6.
 - 5 The input x_4 equals 8.
 - 6 Write the first assignment inside the function.
 - 7 Substitute 2 for x_1 , 4 for x_2 , 6 for x_3 and 8 for x_4 .
 - 8 Evaluate the expression.
 - 9 Write the return expression.
 - 10 Substitute 20 for $total$.
 - 11 Evaluate the return expression.
 - 12 Write the assignment.
 - 13 Substitute 5 for $mean(2,4,6,8)$.

WRITE

```
a x = volumePrism(10, 40)  
height = 10  
area = 40  
volumePrism(10, 40) = height * area  
volumePrism(10, 40) = 10 * 40  
  
volumePrism(10, 40) = 400  
x = volumePrism(10, 40)  
x = 400
```

```
b x = mean(2, 4, 6, 8)  
x1 = 2  
x2 = 4  
x3 = 6  
x4 = 8  
  
total = x1 + x2 + x3 + x4  
total = 2 + 4 + 6 + 8  
  
total = 20  
mean(2, 4, 6, 8) = total / 4  
mean(2, 4, 6, 8) = 20 / 4  
mean(2, 4, 6, 8) = 5  
x = mean(2, 4, 6, 8)  
x = 5
```

20.5.5 Designing an algorithm

- An **algorithm** is a step-by-step set of tasks to solve a particular problem. A program is the implementation of the algorithm.
- Designing an algorithm to solve a problem involves:
 - determining the inputs
 - determining the output required
 - breaking the problem into simple steps.These steps are not code, but an explanation of what the program is required to do at this point.

WORKED EXAMPLE 24

Design an algorithm to solve each of the problems below.

- Determine the inputs.
- Determine the output required.
- Break the problem into simple steps.
 - Convert from a decimal to a percentage.
 - Calculate the total area of a number of squares with the same side length.
 - Calculate the probability of a picking a red marble from a bag of red marbles and blue marbles.

THINK

- Write the input.
 - Write the output required.
 - Write step 1.
- Write the inputs.
 - Write the output required.
 - Write step 1.
 - Write step 2.
- Write the inputs.
 - Write the output required.
 - Write step 1.
 - Write step 2.

WRITE

- decimal
 - percentage
 - Find the percentage from a decimal.
- squares, length
 - area
 - Calculate area of each square.
Calculate total area.
- red, blue
 - probability
 - Find the total number of marbles.
Find the probability of a red marble.

20.5.6 Implementing an algorithm

- Implementing an algorithm in JavaScript involves:
 - designing an algorithm
 - writing function inputs
 - writing JavaScript statements for each step
 - returning the required result (output).

WORKED EXAMPLE 25

Implement an algorithm in JavaScript to solve each of the problems below.

- Convert from a decimal to a percentage.
- Calculate the total area of a number of squares with the same side length.
- Calculate the probability of a picking a red marble from a bag of red marbles and blue marbles.

THINK

- 1 Design an algorithm.
- 2 Write the function input.
- 3 Write step 1.
- 4 Return the required result.

- 1 Design an algorithm.
- 2 Write the function inputs.
- 3 Write step 1.
- 4 Write step 2.
- 5 Return the required result.

- 1 Design an algorithm.
- 2 Write the function inputs.
- 3 Write step 1.
- 4 Write step 2.
- 5 Return the required result.

WRITE

- a See Worked example 24a.

```
function toPercentage(decimal) {  
    // Find percentage from decimal  
    var percentage = 100 * decimal;  
    return percentage;  
}
```

- b See Worked example 24b.

```
function area(number, length) {  
    // Calculate area of each square  
    var squareArea = length * length;  
    // Calculate total area  
    var totalArea = number * squareArea;  
    return totalArea;  
}
```

- c See Worked example 24c.

```
function prob(red, blue) {  
    // Find total number of marbles  
    var marbles = red + blue;  
    // Probability of a red marble  
    var probability = red / marbles;  
    return probability;  
}
```

Exercise 20.5 Algorithms

assessment

Individual pathways

PRACTISE

Questions:
1–9

CONSOLIDATE

Questions:
1, 2a–d, 3, 4a–c, 5a–c, 6–9, 11, 12

MASTER

Questions:
2d–f, 3, 4c, d, 5b–d, 6, 7b, c, 8–14

To answer questions online and to receive **immediate feedback** and **sample responses** for every question, go to your learnON title at www.jacplus.com.au. *Note:* Question numbers may vary slightly.

Fluency

- WE16** i. What is the name of each of the following functions?
ii. What is the output to the console if each function was called?

a.

```
function rectangleArea() {  
    var width = 30;  
    var height = 56;  
    console.log(width * height);  
}
```

b.

```
function outputResults() {  
    var result1 = 1;  
    var result2 = 5;  
    var result3 = 15;  
    var sum = result1 + result2 + result3;  
    console.log(sum / 3);  
}
```

c.

```
function totalTimeMinutes() {  
    var timeHours1 = 16;  
    var timeHours2 = 71;  
    console.log(60 * (timeHours1 + timeHours2) / 2);  
}
```

- WE17** Given the following function names, what is the statement to call each function?

a. callThis

b. divideThenMultiply

c. outputMeanValue

d. outputTotal

e. outputCircleArea

f. calculateMode

- WE18** Are each of the following valid programs? (Answer either true or false.)

a.

```
function defineFirst() {  
    console.log("Output");  
}
```

```
defineFirst();
```

b.

```
function whyCallMe() {  
}
```

```
whyCallMe();
```

c.

```
callLater();
```

```
function callLater() {  
    console.log("Too late now");  
}
```

d.

```
function definedFunction() {  
    console.log("Output second and third");  
}
```

```
console.log("Output first");  
definedFunction();  
definedFunction();
```

- WE20** For each function below, answer the following questions.

i. What is the name of the function?

ii. What is the name of the input?

iii. What is the output to the console if the function was called with the input value 6?

a.

```
function outputTriple(input) {  
    console.log(input + input + input);  
}
```



```

b. function outputArea(s) {
    console.log(s * s);
}
c. function cubeVolume(length) {
    var area = length * length;
    console.log(length * area);
}
d. function twoTimesSquare(a) {
    console.log(2 * a * a);
}

```

5. **WE21** i. How many inputs do the each of following functions have?
 ii. What are the names of the inputs for each function? (Write the inputs in order with no spaces and separated with commas.)

```

a. function threeInputs(x,y,z) {
    console.log(x * x + y * y + z * z);
}
b. function ignoreInputs(ignore1,ignore2) {
    console.log(10);
}
c. function constantOutput(input) {
    console.log("Never changing");
}
d. function sumMany(a0,a1,a2,a3,a4,a5,a6,a7) {
    console.log(a0 + a2 + a3 + a4 + a5 + a6 + a7);
}

```

Understanding

6. **WE19** a. Write a function named `outputBedTime` to output to the console "9:00PM".
 b. Write a statement to call the function.
 c. Combine the function definition and statement into one program.

7. **WE22** What is the output to the console of each of the following programs?

```

a. function outputPowers(x) {
    console.log(x * x * x * x * x);
}
outputPowers(5);
b. function squareArea(side) {
    console.log(side * side);
}
squareArea(40);
c. function oneInput(butIgnored) {
    console.log("No input required");
}
oneInput(23847324);

```

8. **WE23** Manually simulate running the following program. What is the final value stored in the variable `x`?

```

function findArea(width,height) {
    return width * height;
}
var x = findArea(12,5);

```

9. Manually simulate running the following program. What is the final value stored in the variable x ?

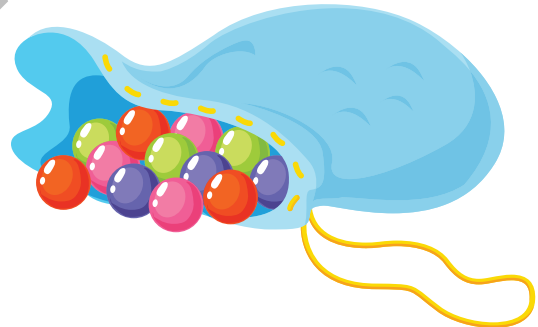
```
function multiply(x1,x2) {
    return x1 * x2;
}
function multiplyAddCombinations(a,b,c) {
    var combination1 = multiply(a,b);
    var combination2 = multiply(b,c);
    var combination3 = multiply(c,a);
    return combination1 + combination2 + combination3;
}
var x = multiplyAddCombinations(2,3,4);
```

10. Manually simulate running the following program. What is the final value stored in the variable x ?

```
function f1() { return 1; }
function f2() { return 1; }
function f3() { return f1() + f2(); }
function f4() { return f2() + f3(); }
function f5() { return f3() + f4(); }
function f6() { return f4() + f5(); }
function f7() { return f5() + f6(); }
function f8() { return f6() + f7(); }
var x = f8();
```

Reasoning

11. **WE24** Design an algorithm to solve each of the problems below.
- Determine the inputs.
 - Determine the output required.
 - Break the problem into simple steps.
 - Convert from a percentage to a decimal.
 - Calculate the probability of picking a red marble from a bag of red, purple, green and pink marbles.



12. **WE25** Implement an algorithm in JavaScript to solve each of the problems below.
- Convert from a percentage to a decimal.
 - Calculate the probability of picking a red marble from a bag of red, purple, green and pink marbles.

Problem solving

13. Write a function named `someFunction` that returns the following results when called.

Statement	y
<code>var y = someFunction(1);</code>	2
<code>var y = someFunction(12);</code>	156
<code>var y = someFunction(6);</code>	42
<code>var y = someFunction(7);</code>	56

15. Extension

Design and implement a program that accepts the inputs a , b and c . The program returns the value x , which solves the mathematical equation. $ax + b = c$.

Reflection

Where do you see applications of algorithms in everyday life?

20.6 Review

Fluency

- Classify the following values as number, string or Boolean.
 - 64
 - 2.34
 - true
 - "Some random text"
 - "true"
 - false
 - 2
 - "-22"
 - 1222.3
 - 86
 - "999"
- Is each of the following a string value? (Answer either true or false.)
 - "A string value"
 - This is a string
 - true
 - 374
 - " 1234"
 - "1234"
 - false
 - value1 + value2
 - 1.234
 - 1.1.1.1.1
 - "3464812.9879879"
- Is each of the following a number value? (Answer either true or false.)
 - "Forty"
 - "40"
 - 2653
 - "0"
 - 0
 - true
 - Eighteen
 - false
 - "238476"
 - 6.22
- Is each of the following a Boolean value? (Answer either true or false.)
 - false
 - true
 - "true"
 - "false"
 - False
 - True
 - true
 - 1
- Variable names:
 - must not start with a number
 - can only contain upper-case and lower case letters, numbers and the underscore character `_`
 - cannot be JavaScript keywords. The following are some JavaScript keywords that should not be used as variable names: `true`, `var`, `false`.Is each of the following a valid variable name? (Answer either true or false.)
 - thisIs0k
 - output_results
 - output results
 - outputResults
 - 1stResult
 - secondResult
 - 3rdResult
 - four%two
 - no good
- Evaluate each of the following JavaScript expressions manually.
 - $2 + 31$
 - $84 - 34$
 - $-14 + 101$
 - "No" + "is a number"
 - "length ten".length
 - $(17 + 3 * (14 - 6) / 2 - 2) / 9$
 - $11 * 7$
 - $18 * 2$
 - $(2 + 31) * 4 / 12$
 - $16 + (6 - 5) * 12$
 - "Remember quot" + "ation marks"
 - $(7 - 5) / 2 + (12 + 16) / 14$
 - ("Join" + "This").length
 - $(12 + 13) / 5 + (4 + 16) / 4$
 - $200 / 40$
 - "Join".length + "This".length

7. Match the JavaScript comments below with the appropriate statements in the programs in the table.

```
// calculate the area of triangle 1
// calculate the area of triangle 2
// triangle 2 has half the height of triangle 1
// set the base of triangle 1 to 16
// triangle 2 has double the base of triangle 1
// set the height of triangle 1 to 10
```

Program	Comment
<code>var height1 = 10;</code>	
<code>var base1 = 16;</code>	
<code>var height2 = height1 / 2;</code>	
<code>var base2 = 2 * base1;</code>	
<code>var area1 = height1 * base1 / 2;</code>	
<code>var area2 = height2 * base2 / 2;</code>	

8. What is output to the console after running each of the following statements?

- `console.log((38 + 12) / (16 - 6));`
- `console.log(("Measure the " + " " + "length").length);`
- `console.log("Console output has no " + "quotation marks");`
- `console.log(("This is g").substring(8, 9));`
- `console.log(("Length is 12").substr(14, 2));`

9. Manually simulate running the following programs. What is the final value stored in the variable p?

- ```
var v = 4 + 96;
var p = v / 4;
```
- ```
var y1 = "heals";
var y2 = "all";
var y3 = "time";
var s = " ";
var p = y3 + s + y1 + s + y2;
```
- ```
var changing = 12;
var changing = 11;
var changing = 1;
var p = changing + changing + changing;
var changing = 2;
var x = changing;
```

10. Manually simulate running the following programs. What will be output to the console?

- ```
var b = 11;
console.log(b * b + b);
```
- ```
var fourteen = "14";
console.log(fourteen.length * 5 + 4);
```
- ```
var x = 9;
var y = 12;
var z = 15;
console.log(x * x + y * y - z * z);
```
- ```
var single = "One";
var double = "two";
var triple = "three";
var three = single + " plus " + double + " equals " + triple;
console.log(three);
```

```
e. var x = "Java";
 var y = "Script";
 console.log(x + y);
```

11. Match the JavaScript statements below with the equivalent R statements in the table.

```
var y = 15;
var t = 10;
var y = 4 + 10;
var t = "10";
```

| R statements | JavaScript statements |
|--------------|-----------------------|
| t <- 10      |                       |
| t <- "10"    |                       |
| y <- 10 + 4  |                       |
| y <- 4 + 11  |                       |

12. i. What is the name of each of the following functions?

ii. What is the output to the console if each function was called?

```
a. function averageWidthScaled() {
 var width1 = 30;
 var width2 = 50;
 var averageWidth = (width1 + width2) / 2;
 var scale = 40;
 console.log(averageWidth * scale);
}
```

```
b. function averageResults() {
 var x1 = 31;
 var x2 = 335;
 var x3 = 1335;
 var x4 = 13;
 var Sum = x1 + x2 + x3 + x4;
 var Average = Sum / 4;
 console.log("Average");
}
```

13. Given the following function names, what is the statement to call each function?

- interestingName
- outputTime
- meanValue

14. Are the following valid programs? (Answer either true or false.)

```
a. function correctOrder() {
 console.log("Working");
}
correctOrder();
```

```
b. function I_do_nothing() {
}
I_do_nothing();
```

```
c. jumpTheGun();
function jumpTheGun() {
 console.log("False start");
}
```

- ```

d. function callMeAgain() {
    console.log("Do it again");
}
console.log("First time");
callMeAgain();
callMeAgain();
callMeAgain();

```
15. i. What is the name of each of the following functions?
 ii. What is the name of the input in each case?
 iii. What is the output to the console if each function was called with the input value 7?
- ```

a. function doubleIt(doubleThis) {
 console.log(doubleThis + 2 * doubleThis - doubleThis);
}
b. function outputCubeVolume(side) {
 console.log(side * side * side);
}

```
16. i. How many inputs do each of the following functions have?  
 ii. What are the names of the inputs in each case? (Write the inputs in order with no spaces and separated with commas.)
- ```

a. function countThem(x,y) {
    console.log(x / x + y / y);
}
b. function useInputs(use1,use2,use3) {
    var total = use1 + use2 + use3;
    console.log("Maybe ignore them");
}

```
17. What is the output to the console of the following program?
- ```

function power3(y) {
 console.log(y * y * y);
}
function power6(x) {
 console.log(power3(x) * power3(x));
}
power6(2);

```
18. Manually simulate running the program below. What is the final value stored in the variable volume?
- ```

function findVolume(width,height,depth) {
    return width * height * depth;
}
var volume = findVolume(3,6,8);

```

Problem solving

1. Manually simulate running the following program. What is the final value stored in the variable x?
- ```

function dice1(a,b,c,d,e,f) {
 return d;
}
function dice2(y1,y2,y3,y4,y5,y6) {
 return y5;
}

```

```

function twoDice(z1,z2,z3,z4,z5,z6) {
 var role1 = dice1(z1,z2,z3,z4,z5,z6);
 var role2 = dice2(z1,z2,z3,z4,z5,z6);
 var total = role1 + role2;
 return total;
}
var x = twoDice(6,5,4,3,2,1);

```

2. Manually simulate running the following program. What is the final value stored in the variable x?

```

function f1() { return 1; }
function f2() { return 1; }
function f3() { return 1; }
function f4() { return f1() + f2() + f3(); }
function f5() { return f2() + f3() + f4(); }
function f6() { return f3() + f4() + f5(); }
function f7() { return f4() + f5() + f6(); }
var x = f7();

```

3. Design an algorithm to solve each of the problems below.

- i. Determine the inputs.
- ii. Determine the output required.
- iii. Break the problem into simple steps.

- a. Take the base to the power of 4 and return the value.
  - b. Find the mean of five numbers.
  - c. Calculate the missing angle of a triangle given the other two angles.
4. Implement an algorithm in JavaScript to solve each of the problems below.
- a. Take the base to the power of 4 and return the value.
  - b. Find the mean of 5 numbers.
  - c. Calculate the missing angle of a triangle given the other two angles.

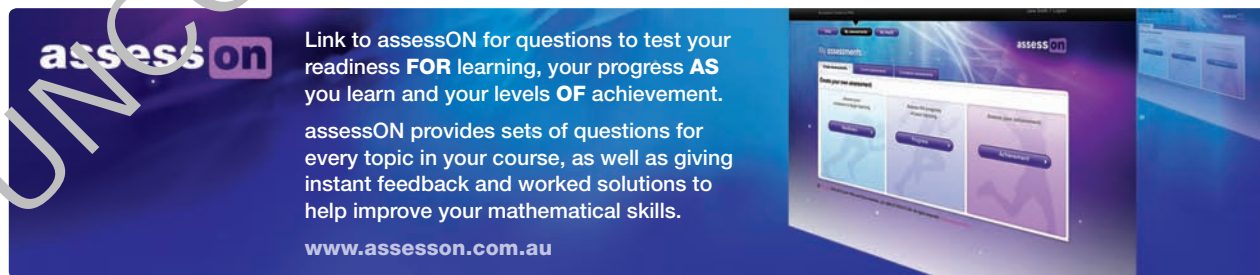
## Language

It is important to learn and be able to use correct mathematical language in order to communicate effectively. Create a summary of the topic using the key terms below. You can present your summary in writing or using a concept map, a poster or technology.

**algorithm**  
**Boolean**  
**character**

**console**  
**called**  
**method**

**number**  
**string**  
**variable**



**assessON** Link to assessON for questions to test your readiness **FOR** learning, your progress **AS** you learn and your levels **OF** achievement.

assessON provides sets of questions for every topic in your course, as well as giving instant feedback and worked solutions to help improve your mathematical skills.

[www.assesson.com.au](http://www.assesson.com.au)



# Answers

## Exercise 20.2 Variables

1. a. Number      b. Number      c. Number      d. String      e. String      f. String  
 g. Boolean      h. String      i. Boolean      j. String      k. Number      l. Number  
 m. String      n. Boolean      o. String      p. String      q. Number      r. String
2. a. False      b. False      c. True      d. True      e. False      f. False      g. False      h. False      i. False  
 j. False      k. False      l. True      m. True      n. True
3. a. True      b. False      c. False      d. True      e. False      f. True      g. True      h. False      i. True  
 j. False      k. True      l. False      m. False      n. True      o. False
4. a. True      b. True      c. False      d. False      e. False      f. False      g. False      h. False      i. False
5. a. True      b. True      c. True      d. True      e. False      f. False      g. True      h. False      i. False  
 j. True      k. True      l. False      m. True      n. True      o. False      p. False      q. True      r. True  
 s. True      t. True      u. True      v. False      w. False

6.

|      | Variable   | Value          | Type    | JavaScript statement          |
|------|------------|----------------|---------|-------------------------------|
| e.g. | a_variable | "text"         | String  | var a_variable = "text";      |
| a    | x          | 10             | Number  | var x = 10;                   |
| b    | nine       | 9              | Number  | var nine = 9;                 |
| c    | aBoolean   | true           | Boolean | var aBoolean = true;          |
| d    | _          | "A string"     | String  | var _ = "A string";           |
| e    | _long_var  | true           | Boolean | var _long_var = true;         |
| f    | areYouTall | false          | Boolean | var areYouTall = false;       |
| g    | about_e    | 2.718281828459 | Number  | var about_e = 2.718281828459; |
| h    | sin_90     | 1              | Number  | var sin_90 = 1;               |
| i    | isThis     | "false"        | String  | var isThis = "false";         |
| j    | a_Boolean  | true           | Boolean | var a_Boolean = true;         |

7. true, false

8. Yes

9. a. var count = 10;      b. var information = "Data";  
 c. var threeSquared = 9;      d. var isThisJavaScript = true;  
 e. var meaningOfLife = 42;      f. var classSize = 25;
10. a. timeInSeconds < 1.0      b. distanceInMiles <= 40  
 c. firstName <= "Ben"      d. middleName <= "Violet"  
 e. numberLeft <= 19      f. littleOverThree <= 3.1

11. a. True      b. True      c. False      d. False      e. True      f. False      g. True

12.

|   | JavaScript assignment statement | Python assignment statement |
|---|---------------------------------|-----------------------------|
| a | var x = 10;                     | x = 10                      |
| b | var nine = 9;                   | nine = 9                    |
| c | var aBoolean = true;            | aBoolean = True             |
| d | var _ = "A string";             | _ = "A string"              |
| e | var _longer_variable = true;    | _longer_variable = True     |
| f | var areYouTall = false;         | areYouTall = False          |
| g | var nearly4 = 3.99;             | nearly4 = 3.99              |
| h | var about_e = 2.718281828459;   | about_e = 2.718281828459    |
| i | var sin_90 = 1;                 | sin_90 = 1                  |
| j | var isThisABoolean = "false";   | isThisABoolean = "false"    |
| k | var is_this_a_Boolean = true;   | is_this_a_Boolean = True    |



- c. i. "Use hands".substr(5,3)  
 ii. "Use hands".substring(5,8)
- d. i. "Search the land".substr(12,3)  
 ii. "Search the land".substring(12,15)
- e. i. "ksanbjhandasdkjh".substr(7,3)  
 ii. "ksanbjhandasdkjh".substring(7,10)
- f. i. "Earth and Mars".substr(6,3)  
 ii. "Earth and Mars".substring(6,9)
8. a. "x"    b. "z"    c. "5"    d. "."
9. a. "MAKE ALL UPPER CASE"    b. "make all lower case"  
 c. "A TITLE"    d. "stop shouting"
10. a. 2    b. 4    c. 0    d. 2    e. 1    f. 3
11. a. `var finalResult = 27 - (16 - 3);`  
 b. `var finalResult = (3 + 23) / (10 + 3);`  
 c. `var finalResult = "Assign" + "ed to finalResult";`  
 d. `var finalResult = "Len".length + "gth".length;`  
 e. `var finalResult = 18%5;`
12. a. `var numberOfCarsEach = 12 / 3;`  
 b. `var totalCash = 20 + 15;`  
 c. `var myTitle = "My Trip".toUpperCase();`
13. a. `var questionResult = 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1;`  
 b. `var questionResult = 2 + 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18;`  
 c. `var questionResult = ("Th" + "e" + "f" + "in" + "al re" + "sult is 2" + "2").length;`
14. a. `1*23456789` or `23456789*1`    b. `9642*87531` or `87531*9642`
15. Many answers are possible. A sample answer is shown.  
`var englishWord = "program";`  
`var wordLength = englishWord.length;`  
`var firstLetter = englishWord.`  
`substr(0, 1);`  
`var restOfWord = englishWord.substring`  
`(1, wordLength);`  
`var pigLatin = restOfWord + firstLetter + "ay";`  
`console.log(pigLatin);`

### Exercise 20.4 Sequences

1.

| Program                                       | Comment                                      |
|-----------------------------------------------|----------------------------------------------|
| <code>var width = 10;</code>                  | <code>// set the width to 10</code>          |
| <code>var height = 15;</code>                 | <code>// set the height to 15</code>         |
| <code>var halfAround = width + height;</code> | <code>// calculate half the perimeter</code> |
| <code>var allAround = 2 * halfAround;</code>  | <code>// calculate the perimeter</code>      |
| <code>var area = width * height;</code>       | <code>// calculate the area</code>           |

2. a. 4    b. 16    c. No quotation marks in console output  
 d. d    e. 18    f. 16

3. a. 10    b. "cat slept"    c. 44

4. a. 105    b. 10    c. 0  
 d. One plus two    e. 1020    f. 20

| C++ statements                              | JavaScript statements                       |
|---------------------------------------------|---------------------------------------------|
| <code>int y = 10 + 3;</code>                | <code>var y = 3 + 10;</code>                |
| <code>int x = 10 + 4;</code>                | <code>var x = 10 + 4;</code>                |
| <code>int x = 3 + 10;</code>                | <code>var x = 10 + 3;</code>                |
| <code>int x = 10 * 4;</code>                | <code>var x = 10 * 4;</code>                |
| <code>int aVariable = 9 * 8 * 7 * 6;</code> | <code>var aVariable = 6 * 8 * 7 * 9;</code> |

6. a. `var nearly_e = 2 + 7 / 10 + 1 / 100 + 8 / 1000; // nearly_e = 2.718`  
 b. `var marathonKm = 26 * 1.6; // marathonKm = 41.6;`  
 c. `var secondsPerDay = 60 * 60 * 24; // secondsPerDay = 86400`
7. a. `console.log((3 + 41) / (9 + 2));`      b. `console.log("putter".substr(1,5));`  
 c. `console.log(2 + 3 - 5);`      d. `console.log((2 * 12)/(8 - 2));`  
 e. `console.log("life".substring(1,3));`
8. a. `var side1 = 3 * 4; // side1 = 12`  
     `var side2 = side1 / 2; // side2 = 6`  
     `var area = side1 * side2; // area = 72`  
 b. `var a = 5 + 2; // a = 7`  
     `var b = a + a; // b = 14`  
     `var c = b + b; // c = 28`  
     `var delta = 4 * a * c - b * b;`  
     `// delta = 588`  
 c. `var reassign = 12 + 18; // reassign = 30`  
     `var reassign = 1 + 41; // reassign = 42`  
     `var result = 3 * reassign; // result = 126`
9. `var r = 8;`  
     `var r_less_5 = r - 5;`
10. a. `int minutes = 34;`  
        `int seconds = minutes * 60;`  
 b. `var hours = 3;`  
     `var minutes = hours * 60;`
11. a. TypeScript      b. R      c. PHP      d. JavaScript      e. TypeScript      f. JavaScript
12. a and b

|   |                                               |   |                                               |
|---|-----------------------------------------------|---|-----------------------------------------------|
| A | <code>var e = 5;</code>                       | A | <code>var e = 5;</code>                       |
| B | <code>e = (e + 7) * 2;</code>                 | C | <code>e = e / e;</code>                       |
| C | <code>e = e / e; console.log(e);</code>       | B | <code>e = (e + 7) * 2; console.log(e);</code> |
| B | <code>e = (e + 7) * 2;</code>                 | B | <code>e = (e + 7) * 2;</code>                 |
| A | <code>var e = 5;</code>                       | C | <code>e = e / e;</code>                       |
| C | <code>e = e / e; console.log(e);</code>       | A | <code>var e = 5; console.log(e);</code>       |
| C | <code>e = e / e;</code>                       | C | <code>e = e / e;</code>                       |
| A | <code>var e = 5;</code>                       | B | <code>e = (e + 7) * 2;</code>                 |
| B | <code>e = (e + 7) * 2; console.log(e);</code> | A | <code>var e = 5; console.log(e);</code>       |

|   |    |   |    |
|---|----|---|----|
| A | 1  | A | 16 |
| B |    | C |    |
| C |    | B |    |
| B | 1  | B | 5  |
| A |    | C |    |
| C |    | A |    |
| C | 24 | C | 5  |
| A |    | B |    |
| B |    | A |    |

```

13. var width = 30;
 var length = 20;
 var perimeter = 2 * width + 2 * length;
14. var width = 12;
 var height = 9;
 var triangleArea = height * width / 2;
 var squareArea = width * width;
 var totalArea = squareArea + triangleArea;
15. var x = 3;
 x = x + 9;
 x = x * x;
 x = x - 96;
 x = 10 * x;
 x = x / 24;
 x = x - 10;
 x = 110 - x;
16. var x = 9;
 x = x / 3;
 x = x + 1;
 x = x / 2;
 x = x + 8;
 x = x / 5;
 x = x + 7;
 x = x - 6;
 x = 4 - x;

```

### Exercise 20.5 Algorithms

1. a. i. rectangleArea                   ii. 1680  
    b. i. outputResults               ii. 7  
    c. i. totalTimeMinutes           ii. 2610
2. a. callThis();                   b. divideThenMultiply();           c. outputMeanValue();  
    d. outputTotal();               e. outputCircleArea();           f. calculateMode();
3. a. True     b. True     c. False     d. True
4. a. i. outputTriple               ii. input                           iii. 18  
    b. i. outputArea               ii. s                               iii. 36  
    c. i. cubeVolume               ii. length                       iii. 216  
    d. i. twTimeSquare           ii. a                               iii. 72
5. a. i. 2                           ii. x, y, z  
    b. i. 2                           ii. ignore1, ignore2  
    c. i. 1                           ii. input  
    d. i. 8                           ii. a0, a1, a2, a3, a4, a5, a6, a7
6. a. function outputBedTime() {  
       console.log("9:00PM");  
    }  
    b. outputBedTime();  
    c. function outputBedTime() {  
       console.log("9:00PM");  
    }  
    outputBedTime();

7. a. 243

b. 1600

c. No input required

8. 60

9. 26

10. 21

11. a. i. percentage

ii. decimal

iii. Find percentage from decimal.

b. i. red, purple, green, pink

ii. probability

1. Find total number of marbles.

2. Find probability of picking a red marble.

```
12. a. function toDecimal(percentage) {
 var decimal = percentage / 100;
 return decimal;
}
```

```
b. function calculateProbability(red, purple, green, pink) {
 // Find total number of marbles
 var marbles = red + purple + green + pink;
 // Probability of a red marble
 var probability = red / marbles;
 return probability;
}
```

```
13. function someFunction(x) {
 return x * x + x;
}
```

```
14. function xSolve(a, b, c) {
 // Algebraically solve for x
 // a * x + b = c
 // a * x + b - b = c - b
 // a * x = c - b
 // a * x / a = (c - b) / a
 // x = (c - b) / a
 // Calculate x given a, b and c
 return (c - b) / a;
}
```

## 20.6 Review

- |              |                   |           |           |            |           |
|--------------|-------------------|-----------|-----------|------------|-----------|
| 1. a. Number | b. Number         | c. Number | d. String | e. Boolean | f. Number |
| g. String    | h. Number         | i. String | j. String | k. Boolean |           |
| 2. a. True   | b. False          | c. False  | d. False  | e. True    | f. True   |
| g. False     | h. False          | i. False  | j. False  | k. True    |           |
| 3. a. False  | b. False          | c. True   | d. False  | e. True    | f. False  |
| g. False     | h. False          | i. False  | j. True   |            |           |
| 4. a. True   | b. True           | c. False  | d. False  | e. False   | f. False  |
| g. True      | h. False          |           |           |            |           |
| 5. a. True   | b. True           | c. False  | d. True   | e. False   | f. True   |
| g. False     | h. False          | i. False  |           |            |           |
| 6. a. 33     | b. 50             |           |           |            |           |
| c. 87        | d. "Not a number" |           |           |            |           |
| e. 10        | f. 3              |           |           |            |           |
| g. 77        | h. 36             |           |           |            |           |

- i. 11
- j. 28
- k. "Remember quotation marks"
- l. 3
- m. 8
- n. 10
- o. 5
- p. 8

7.

| Program                                       | Comment                                            |
|-----------------------------------------------|----------------------------------------------------|
| <code>var height1 = 10;</code>                | <code>// set the height of triangle 1 to 10</code> |
| <code>var base1 = 16;</code>                  | <code>// set the base of triangle 1 to 16</code>   |
| <code>var height2 = height1 / 2;</code>       | <code>// triangle 2 has half the height</code>     |
| <code>var base2 = 2 * base1;</code>           | <code>// triangle 2 has double the base</code>     |
| <code>var area1 = height1 * base1 / 2;</code> | <code>// calculate the area of triangle 1</code>   |
| <code>var area2 = height2 * base2 / 2;</code> | <code>// calculate the area of triangle 2</code>   |

- 8. a. 5
- b. 18
- c. Console output has no quotation marks
- d. g
- e. 12
- 9. a. 25
- b. "time heals all"
- c. 3
- 10. a. 132
- b. 14
- c. 0
- d. One plus two equals three
- e. JavaScript

11.

| R statements                | JavaScript statements        |
|-----------------------------|------------------------------|
| <code>t &lt;- 10</code>     | <code>var t = 10;</code>     |
| <code>t &lt;- "10"</code>   | <code>var t = "10";</code>   |
| <code>y &lt;- 10 + 4</code> | <code>var y = 4 + 10;</code> |
| <code>y &lt;- 4 + 11</code> | <code>var y = 15;</code>     |

- 12. a. i. `averageWidthScaled`
- ii. 1600
- b. i. `averageResults`
- ii. `Average`
- 13. a. `interestingName()`
- b. `outputTime()`
- c. `meanValue()`
- 14. a. True
- b. True
- c. False
- d. True
- 15. a. i. `doubleIt`
- ii. `doubleThis`
- iii. 14
- b. i. `outputCubeVolume`
- ii. `side`
- iii. 343
- 16. a. i. 2
- ii. `x, y`
- b. i. 3
- ii. `use1, use2, use3`

- 17. 64
- 18. 14
- 19. 7
- 20. 17

- 21. a. i. base
- ii. value
- iii. 1) Multiple base by base by base by base.
- b. i. `x1, x2, x3, x4, x5`
- ii. mean
- iii. 1) Find the total of `x1, x2, x3, x4, x5`.
- iv. 2) Divide the total by 5.
- c. i. `angle1, angle2`
- ii. `angle3`
- iii. 1) Find the total of known angles.
- iv. 2) Subtract the above total from 180.

```
22.a. function toPower4(base) {
 // Multiple base by base by base by base
 var value = base * base * base * base;
 return value;
}
b. function findMean(x1,x2,x3,x4,x5) {
 // Find the total of x1,x2,x3,x4,x5
 var total = x1 + x2 + x3 + x4 + x5;
 // Divide the total by 5
 var mean = total / 5;
 return mean;
}
c. function findMissingAngle(angle1,angle2) {
 // Find the total of known angles
 var knownTotal = angle1 + angle2;
 // Subtract the above total from 180
 var angle3 = 180 - knownTotal;
 return angle3;
}
```

UNCORRECTED PAGE PROOFS