

### 1. Alert command

1.1. use **alert** to show a message

In **JavaScript** the **alert** command is used to display messages. For example, the code `alert('Hello')` displays the message: "Hello"

Notice that `alert` is followed by (, then the message inside quotes ', then ).

The code editor contains an alert command. Do the following to complete your 1<sup>st</sup> task:

1. On line 1 in the code editor, replace ??? with your name.
2. Click the button to run the code.

1.2. use `\n` to put text on a new line

`\n` is a special character called a **newline**. It splits a message into multiple lines.

1. Put `\n` between `Hello!` and `My` in the **alert** message. The `\` key is above ENTER on US keyboards, and next to Z for UK keyboards.
2. Click to run the code and look at the message.

When typing the newline character (`\n`), use backslash ( `\` ) **NOT** forward slash ( `/` ).

1.3. add another alert command

For this task:

1. On l#1 add a 2<sup>nd</sup> `alert` command.
2. Make the 2<sup>nd</sup> **alert** message display the names of your top 2 favorite websites on separate lines.
3. When you are done, click .

1.4. fix bugs in an alert command

It is easy to put mistakes in your code that make it fail. These mistakes are called **bugs**.

1. the code and read the error message that appears in the output console. The error messages the **web browser** displays are confusing for beginners. But don't worry, you'll understand them over time.
2. Add the missing character on line 1.
3. Add the missing character on line 2.
4. Re-click to complete the task.

1.5.

Review Quiz Questions:

1. What is the name of the JavaScript command used to display a message box?
2. What special character splits **alert** messages into multiple lines?
3. Which character follows the word **alert** in an alert command?
4. Which term is another word for errors in a computer program?

### 2. Variables & prompts

2.1. use the **prompt** command

The **prompt** command is used to ask questions.

1. On line 1, change the **alert** to **prompt**.
2. Click to run the code.

2.2. save data in a **variable**

To make code more useful, we use **variables** to store data for later use.

Line 1 in the code creates a variable called `message` and stores the message **Coding is FUN** inside it. The keyword **var** is put at the start of lines that create variables.

Line 3 displays the value stored in the **message** variable with an **alert** box.

1. Click to run the code.
2. Change the text stored in the **message** variable.
3. the code again.

2.3. store user responses with **variables**

Line 1 in the example asks the user's name and stores the answer in a **variable** called: `name`

Line 2 displays the name in an **alert** box.

1. Make line 1 of your code store the user's **age** in a **variable** called: `age`
2. On line 2, display the value of the `age` **variable** in an **alert** box.
3. On line 4, use another **prompt** to ask the user's favorite color, and store the value in a variable called `color` (coders normally use American spelling for variable names).
4. On line 5, display the value of `color` in an **alert** box.
5. Click and enter your age and favorite color then click OK.

We set the background of Code Avengers to the color you type in the 2<sup>nd</sup> prompt. **Click here for a list of accepted colors.**

2.4. store user responses with **variables**

Line 2 in the example says hello to the user. If the user's name is **Mike** the message will say: Hello Mike

1. Use `+` to add the text `'Your age is '` before the **age** variable in the 1<sup>st</sup> alert.  
The message should say **Your age is 27** if the user type 27 as their age.
2. Use `+` to add the text `' is a nice color'` after the **color** variable in the 2<sup>nd</sup> alert.  
The message should say **blue is a nice color** if the user types **blue**.
3. Click and enter your age and favorite color then click OK.

**Click here for the color list.**

2.5.

Review Quiz Questions:

1. Which of the following keywords is used to create a variable?
2. Which of the following creates a variable `x` that contains the text `hello`?
3. What does the alert message say when the following code runs:
4. What does the alert message say when the following code runs:

### 3. Robot challenge

### 3.1. move the robot forward

Let's take a short break and do the **Robot Challenge!**

The command `robot.forward()` moves the robot forward 1 square. To move forward 3 squares with a single command use `robot.forward(3)`.

1. Write code to move the robot to the square marked **X**.
2. Click to run your code.

### 3.2. turn the robot right

To turn right 90 degrees use: `robot.right()` Put each command on a separate line and don't forget the `()`.

1. Write code to move the robot to the **X**. It is facing **up** at the start.
2. Click to run your code.

### 3.3. turn the robot left

To turn left 90 degrees use: `robot.left()`

1. Write code to move the robot to the **X**.
2. Click to run your code.

### 3.4. turn the robot 180 degrees

`robot.right()` turns the robot 90 degrees. To turn right 180 degrees repeat `robot.right()` twice, or use: `robot.right(2)`

1. Write code to move the robot to the **X**.
2. Click to run your code.

### 3.5. move the robot through a path

1. Write code to move the robot to the **X**
2. Click to run your code

## 4. If statements

### 4.1. run an if statement

In a quiz game your code must check if the user answers correctly. This is done using an **if statement** as shown in the code.

Line 1 asks a math question and stores the response in a variable called: `guess`

Lines 3-4 are an **if statement**. The alert on line 4 is displayed only if the condition on line 3 is true. The message **correct** is displayed only **if** the **guess** variable is equal to **50**.

Test the code by running it twice:

1. Click and type the correct answer.
2. Click and type the incorrect answer.

### 4.2. run an if else statement

If statements use the **else** keyword to specify code that runs when the condition is false.

The message on line 4 is displayed if the user gets the answer correct. Otherwise, the message on line 6 is shown.

In coding `==` is different to `=`.

Single equals `=` is used to **store values** in variables, as on line 1. Double equals `==` can check if a variable **is equal to** a value, as on line 3.

1. Click and type the correct answer.
2. Click and type the incorrect answer.

### 4.3. fix code layout

It is important to lay out code nicely, so it is easy to understand. The top code works correctly but is hard to read.

1. Make the layout of the top code **exactly** the same as the example code. Use the **tab** key to move a line right 2 spaces.
2. Click and enter the **correct** answer to the question.

### 4.4. fix if statement bugs

It is important to practice fixing **bugs** in code.

Remember to use `==` when checking if a variable **is equal to** a value. Also, the condition that follows an **if statement** must be surrounded by `()`.

The keyword **var** is used once for each variable. So **var** is used on line 1 but not line 8.

The code in the editor is missing some characters:

1. your code and read the error messages.
2. Fix the bug on lines 3-6.
3. Fix the bug on lines 10-13.
4. your code and enter the correct answers.
5. your code and enter incorrect answers.

### 4.5.

Review Quiz Questions:

1. What symbol is used to store a value in a variable?
2. Which characters always surround the condition in an if statement?
3. What symbol means **is equal to**?
4. What word should replace the ??? on line 5?

## 5. Number variables

### 5.1. create a number variable

**Variables** can store **numbers** as well as text. For example, the code on line 1 creates a **variable** with a value of **0**.

Variable values can be changed. Line 4 changes the value of **x** to **10**.

1. On line 1, set the initial value of the variable to **10**.
2. On line 4, change the value of the variable to **20**.
3. Click to run the code.

### 5.2. increase a variable value

Quiz games need a variable that stores the score. The score variable should increase each time a question is answered correctly.

Line 1 creates a score variable. Line 4 increases the value of the score variable by 1.

1. Change lines 4 & 7 to increase the score variable by **5**.
2. Add code on line 10 to increase the score by 5 for the 3<sup>rd</sup> time.
3. On line 11 show the final value of score with an alert.
4. Click to run the code.

### 5.3. keep score in a quiz

The code in the editor keeps the users score. Lines 7 and 16 add 1 to the score if the user gets the questions correct. Line 21 displays the final score.

Notice that `{` and `}` are used on lines 5, 8, 10, 14, 17 & 19. They must surround 2 or more lines of code that run when an **if statement** condition is true.

1. Change question 1 to  $4 \times 4$  and update the answer in the **if statement** condition.
2. Change question 2 to  $5 \times 5$  and update the answer.
3. the code and type the **correct** answers for both questions.
4. the code and type **incorrect** answers for both questions.

#### 5.4. create multi-choice questions

This code asks a multi-choice question. For this task, add a 2<sup>nd</sup> question.

1. On line 12, add a prompt to ask **What is a firewall?**  
The options are **1. Antivirus program, 2. Internet blocker, 3. Internet filter** and **4. Internet logger**.
2. Add an **if statement** similar to lines 5-10, that checks for the correct answer: **3**.
3. At the end of the code, display the final score in an **alert** box.
4. the code and type the **correct** answers for both questions.
5. the code and type **incorrect** answers for both questions.
6. **(Optional)** Add your own questions and test your quiz on your friends.

#### 5.5.

Review Quiz Questions:

1. Which of the following creates a variable **x** with a value of 3.
2. Which of the following increases the value of **x** by 1:
3. What is the value of **x** when the following 2 lines of code are run:
4. What is the value of **x** when the following 2 lines of code are run: