Functions

Quite often we need to perform a similar action in many places of the script.

For example, we need to show a nice-looking message when a visitor logs in, logs out and maybe somewhere else.

Functions are the main "building blocks" of the program. They allow the code to be called many times without repetition.

We've already seen examples of built-in functions, like `alert(message)`, `prompt(message, default)` and `confirm(question)`. But we can create functions of our own as well.

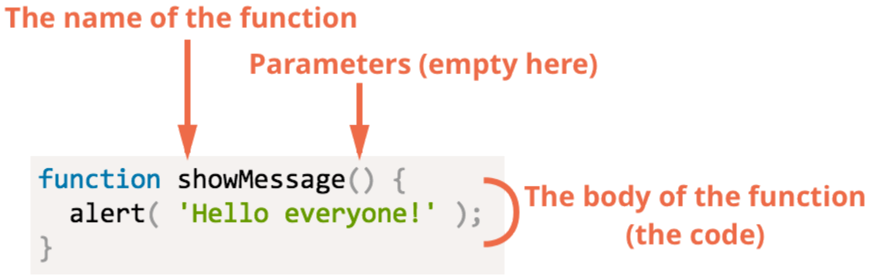
Function Declaration

To create a function we can use a *function declaration*.

It looks like this:

```
function showMessage() {
  alert( 'Hello everyone!' );
}
```

The `function` keyword goes first, then goes the *name of the function*, then a list of *parameters* in the brackets (empty in the example above) and finally the code of the function, also named "the function body".



```
function showMessage() {
  alert( 'Hello everyone!' );
}
```

Our new function can be called by its name: `showMessage()`.

For instance:

```
function showMessage() {
  alert( 'Hello everyone!' );
}

showMessage();
showMessage();
```

The call `showMessage()` executes the code of the function. Here we will see the message two times.

This example clearly demonstrates one of the main purposes of functions: to evade code duplication.

If we ever need to change the message or the way it is shown – it's enough to modify the code in one place: the function which outputs it.

Local variables

A variable declared inside a function is only visible inside that function.

For example:

```
function showMessage() {
  let message = "Hello, I'm JavaScript!"; // local variable

  alert( message );
}

showMessage(); // Hello, I'm JavaScript!

alert( message ); // <-- Error! The variable is local to the function
```

Outer variables

A function can access an outer variable as well, for example:

```


let userName = 'John';

function showMessage() {
  let message = 'Hello, ' + userName;
  alert(message);
}

showMessage(); // Hello, John
```

The function has full access to the outer variable. It can modify it as well.

For instance:

```
let userName = 'John';

function showMessage() {
  userName = "Bob"; // (1) changed the outer variable

  let message = 'Hello, ' + userName;
  alert(message);
}

alert( userName ); // John before the function call

showMessage();

alert( userName ); // Bob, the value was modified by the function
```

The outer variable is only used if there's no local one. So an occasional modification may happen if we forget `let`.

If a same-named variable is declared inside the function then it *shadows* the outer one. For instance, in the code below the function uses the local `userName`, the outer one is ignored:

```
let userName = 'John';

function showMessage() {
  let userName = "Bob"; // declare a local variable

  let message = 'Hello, ' + userName; // Bob
  alert(message);
}

// the function will create and use it's own userName
showMessage();

alert( userName ); // John, unchanged, the function did not access the outer variable
```

Global variables

Variables declared outside of any function, such as the outer `userName` in the code above, are called *global*.

Global variables are visible from any function (unless shadowed by locals).

Usually, a function declares all variables specific to its task, and global variables only store project-level data, so important that it really must be seen from anywhere. Modern code has few or no globals, most variables reside in their functions.

Questions and Exercises

1. Create a program that asks a user for 2 numbers. Make a function that will add and display the sum of these numbers.

   ```
   add();

   product();

   difference();

   divide();



   function add()

   {

    let num1=+prompt("Enter number 1");

    let num2=+prompt("Enter number 2");

    let sum=num1+num2;

    alert("The sum is " + sum);

   }
   ```

```
function product()
{
 let num1=+prompt("Enter number 1");
 let num2=+prompt("Enter number 2");
 let product=num1*num2;
 alert("The product is " + product);
}
function difference()
{
 let num1=+prompt("Enter number 1");
 let num2=+prompt("Enter number 2");
 let diff=num1-num2;
 alert("The difference is " + diff);
}
function divide()
{
 let num1=+prompt("Enter number 1");
 let num2=+prompt("Enter number 2");
 let quotient=num1/num2;
 alert("The sum is " + quotient);
}
```

2. Add 3 new functions to the last program to multiply, divide and subtract the two numbers. Display these results from within each function.

```
let num1=0;
let num2=0;
```

```
add();
product();
difference();
divide();


function add()
{
 num1=+prompt("Enter number 1");
 num2=+prompt("Enter number 2");

 let sum=num1+num2;
 alert("The sum is " + sum);
}
function product()
{
 let product=num1*num2;
 alert("The product is " + product);
}
function difference()
{
 let diff=num1-num2;
 alert("The difference is " + diff);
}
function divide()
{
 let quotient=num1/num2;
 alert("The sum is " + quotient);

}
```

3. Create a program that asks a user if they would like to convert a number from feet to inches or vice versa. Then ask for the number and do the conversion. Create 2 functions. One for conversion from feet to inches and one from inches to feet. Both functions will also display the final result.

```
let number=0;


conversion();


function conversion()

{

 let type=+prompt("Do you want to convert between 1. feet to inches or 2. inches to feet?")

 number=+prompt("What is the number to convert?")

 if(type==1)

   feetToInches();

 else if(type==2)

   inchesToFeet();

}


function feetToInches()

{
```

4. Ask for 2 numbers and then a letter. Create 3 functions that will be called that determine if the first number is between 1 and 5, the second determines if the second number entered is between 1 and 10.5 and the last determines if the letter is between A and D.

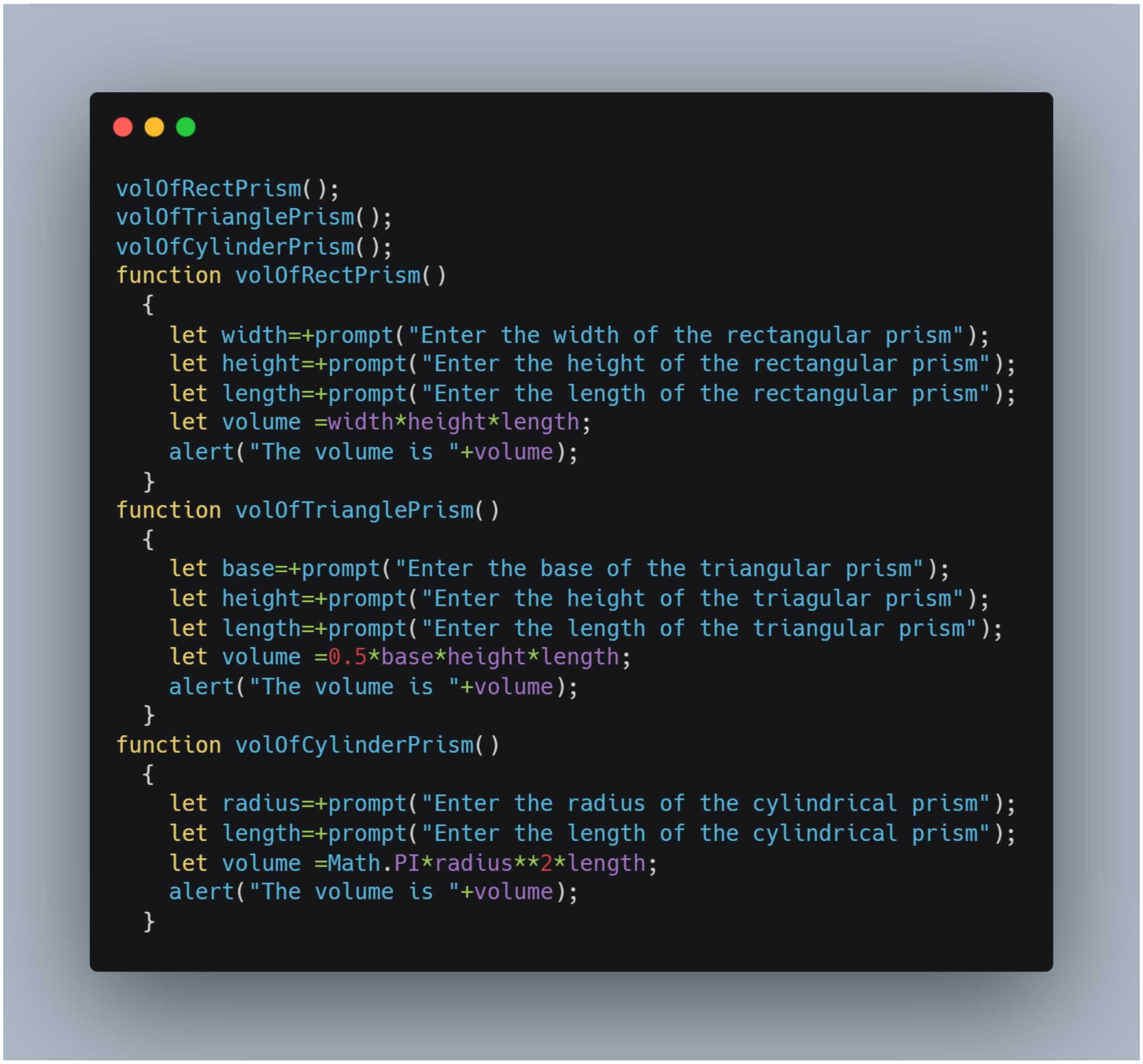
```
between1and5()
```

```
between1and105();
betweenAandD();

function between1and5()
{
 let number=+prompt("Enter a number");
 if(number>=1 && number<=5)
 {alert("The number is between1 and 5");}
 else
 {alert("The number is NOT between1 and 5");}
}
function between1and105(){
 let number=+prompt("Enter a number");
 if(number>=1 && number<=10.5)
 {alert("The number is between1 and 10.5");}
 else
 {alert("The number is NOT between1 and 10.5");}
}
```

5. Create a guessing game where a user tries to guess the computer number. The computer will generate a random number between 1 and 10. Give the user 3 attempts at guessing the number. Use at least 3 functions in the code.

6. Create a program to find the volume of 3 types of prisms i.e. 1. rectangle, 2. triangle and 3. cylinder. Ask the user which one they'd like to calculate the volume of and then call the appropriate function to gather that measurements and display the volume.



```
volOfRectPrism();
volOfTrianglePrism();
volOfCylinderPrism();
function volOfRectPrism()
  {
    let width=+prompt("Enter the width of the rectangular prism");
    let height=+prompt("Enter the height of the rectangular prism");
    let length=+prompt("Enter the length of the rectangular prism");
    let volume =width*height*length;
    alert("The volume is "+volume);
  }
function volOfTrianglePrism()
  {
    let base=+prompt("Enter the base of the triangular prism");
    let height=+prompt("Enter the height of the triagular prism");
    let length=+prompt("Enter the length of the triangular prism");
    let volume =0.5*base*height*length;
    alert("The volume is "+volume);
  }
function volOfCylinderPrism()
  {
    let radius=+prompt("Enter the radius of the cylindrical prism");
    let length=+prompt("Enter the length of the cylindrical prism");
    let volume =Math.PI*radius**2*length;
    alert("The volume is "+volume);
  }
```

7. Create a program to calculate the cost of buying items at a store. Make a function to ask for the item and quantity and a function to calculate the taxes and final total. Use a loop to allow the user to enter the cost and quantity of 3 items. Create one final function to display the subtotal, taxes and final total.

8. Create a dice rolling game where the user places a bet and a guess as to what the sum of two dice rolls will be. The user starts off with a 100. They can play until they lose all their money. They bet only an amount they have. Then they make a guess between 2 and 12. Then they roll and if they win they get back twice their bet. Make a function to get and validate the bet, one to roll and display the dice and their sum and one to check if they win and recalculate their new winnings.

9. Create a program that asks a user for a number n. DIsplay a menu of options of what to do with n. These are as follows: 1. Show all even numbers up to n, 2. show all odd numbers up to n 3. calculate the sum of

all consecutive numbers to n and 4. display the squares and cubes of all numbers up to n. Make functions to solve all these problems.

10. Make a simple 3 question quiz. Create and test for correctness of each question in their own function. At the end display the results.