Variables

Must begin with a letter (A-Za-z), or an underscore (_)
May be followed by other letters (A-Za-z), numbers(0-9) or underscores (_)

Use var when declaring a variable, ie: using it for the first time.

name used for the first time, so use var because we are declaring it

name used for the second time, don’t use var

name used for the third time, don’t use var
## Common Arithmetic Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Adds two numbers</td>
<td>5 + 6</td>
<td>11</td>
</tr>
<tr>
<td>+</td>
<td>Joins two strings</td>
<td>“Abc” + “def”</td>
<td>“Abcdef”</td>
</tr>
<tr>
<td>+</td>
<td>Joins a number &amp; a string</td>
<td>“Abc” + 1</td>
<td>“Abc1”</td>
</tr>
<tr>
<td>-</td>
<td>Subtract two numbers</td>
<td>10 - 5</td>
<td>5</td>
</tr>
<tr>
<td>/</td>
<td>Divide two numbers</td>
<td>10/4</td>
<td>2.5</td>
</tr>
<tr>
<td>%</td>
<td>Get the remainder</td>
<td>10/4</td>
<td>2</td>
</tr>
<tr>
<td>--</td>
<td>Reduce a value by 1</td>
<td>x--</td>
<td>x = x - 1</td>
</tr>
<tr>
<td>++</td>
<td>Increase a value by 1</td>
<td>x++</td>
<td>x = x + 1</td>
</tr>
</tbody>
</table>

Note: + is overloaded, so it does different things.
# Common Relational/Logical Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
</table>
| >=       | Checks if the first is greater or equal than the second | 7 >= 7  
6 >= 7 | true  
false |
| >        | Checks if first is greater than second | 4 > 5  
“b” > “a” | false  
true |
| <=       | Checks if the first is lesser or equal to the second | 7 <= 7  
8 <= 7 | true  
false |
| <        | Checks if first is lesser than the second | 4 < 5  
“a” < “b” | true  
true |
| ==       | Checks if two things are equal | “a” == “b” | false |
| !=       | Checks if two things are not equal | “a” != “b” | true |
Conditional statements (if and else)

The **first** if or else if statement that is true gets executed, others get skipped.

If **none** of the if or else if statements are true, the else statement runs.

```javascript
if (fruit == "Apple"){
    document.write("An apple a day keeps the doctor away!");
}
else if (fruit == "Orange"){
    document.write("Nice! Oranges have a lot of Vitamin C");
}
else if (fruit == "Banana"){
    document.write("I like bananas too!");
}
else{
    document.write("Go ahead, all fruits are good for you!");
}
```
# Logical operators

Let $a = 5$, and $b = 10$ for the examples

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
</table>
| &&       | Checks if both statements are true | $(a == 5 && b == 10)$  
$(a == 5 && b < 10)$ | true  
false |
| ||       | Checks if at least one statement is true | $(a == 4 || b == 18)$  
$(a == 5 || b == 123)$  
$(a == 5 || !b == 10)$ | false  
true  
true |
| !        | Changes a true to false and false to true | !(a == 5)  
!(b == 4)  
(a == 4 || !(b == 123)) | false  
true  
true |
For and While Loops

Loops allow you to run code multiple times based on some condition

In general, all loops have three things in common:

- There is a variable that you are checking in the loop, that you have declared before you started the loop.
- You continue running the loop only if the variable meets some condition.
- The variable changes its value over time.
For loop

declare var i

set i to 0 at the start

keep running if i is less than 100

increase i by 1 each time

print out the value of i

This loop starts with i at value zero, then increases it by 1 each time, until it is 100. It also prints the value of i each time, so it prints the numbers 0 through 99 and stops after 99 because it is not lesser than 100.

change the starting value, the condition to keep running, and the amount i changes by each time, to make the for loop do something else
While loop

- Declare var x and set it to 1
- Keep running if x is less than 1000
- Make x be twice its value each time (double the value of x)

This loop starts with x at value 1, then doubles its value each time, until it is no longer less than 1000.

Change the starting value, the condition to keep running, and the amount i changes by each time, to make the while loop do something else.
Functions

Functions allow you to **write code once** and but **use it many times**

**Define a function calculateTip**

It takes one **parameter**

```javascript
function calculateTip( total ){
  return 0.15 * total;
}
```

- `calculateTip(12.7);` returns **1.905**
- `calculateTip(10);` returns **1.5**
- `calculateTip(20);` returns **3**
Functions continued

Functions can be more complicated, allowing you to save a lot of time as compared to typing the same code again and again.

Define a function `bigger_number`

It takes two parameters

```
function bigger_number(number1, number2){
  if (number1 > number2){
    return number1;
  } else{
    return number2;
  }
}
```

Returns first number if first is bigger

Otherwise, returns second number

```
bigger_number(5,6);
bigger_number(123,45);
```

Returns 6
Returns 123