

# Control Flow

Control flow is the order in which the computer executes statements in a script.

In javascript, we read the code starting from the first line till the last line unless there is some instructions or statements that changes the control flow, such as **conditionals** and **loops**.

# Concept : 'If'

If the value of a boolean expression (condition) is true, do something

eg: If today is Saturday, then I will eat ice-cream

```
if (condition is true) then do stuff in block { }
```

```
if (today == "Saturday") {  
    document.write("Yum, ice-cream!");  
};
```

# Concept : 'If-else'

If the value of a boolean expression is true, do something. Otherwise, do another thing.

eg:

If today is Saturday, then I will eat ice-cream.  
Else, I will eat cookies.

```
if (today == "Saturday") {  
    document.write("Yum, ice-cream!");  
}  
else {  
    document.write("Cookies!");  
};
```

if	(condition is true)
then do stuff in block{ }	
else	(condition is not true)
then do stuff in other block { }	

# Concept : 'If-else if-else'

Add another different condition to test if the first condition **false**.

Eg. If today is Saturday, then I will eat ice-cream.  
Else if today is Monday, I will eat fried rice.  
Else, I will eat cookies.

```
if (today == "Saturday") {  
    document.write("Yum, ice-cream!");  
}  
else if (today == "Monday") {  
    document.write("Fried rice!");  
}  
else {  
    document.write("Cookies!");  
};
```

if	(condition is true)
----	---------------------

then do stuff in block{ }
---------------------------

else if	(this condition is true)
---------	--------------------------

then do stuff in block{ }
---------------------------

else if none of above conditions are true
----------------------------------------------

then do stuff in other block { }
-------------------------------------

# Ordering matters

If a condition is fulfilled, it will run the contents of { ... } and stop checking the other conditions.

So, your conditions should start with the **strictest** one.

How can you fix the code below?

```
let cookies = 10;  
if (cookies > 5) {  
    alert("Eat a cookie");  
} else if (cookies > 8) {  
    alert("Share it around");  
} else {  
    alert("buy more cookies");  
};
```

# Truthy and Falsy

What if your condition is not a boolean?

```
const myString = 'hello';  
  
if (myString) {  
  console.log('hi')  
}
```

# Truthy and Falsy values

A **falsey** value is a value that is considered **false** when encountered in a Boolean context.

```
if (false) //The keyword false
if (null) //the absence of any value
if (undefined) //no assigned value
if (0) //number zero
if (NaN) //not a number
if ("")
if ("")
if (``) } // an empty string or
          // length of the string is zero
```

\*All values are **truthy** unless they are defined as **falsey**

# IF-ELSE exercise :

## FizzBuzz