CONDITIONALS & FUNCTIONS

CONDITIONALS

```
if (condition) {
    // do something
}
```

```
if (condition) {
    // do something
} else {
    // do something else
}
```

```
if (condition) {
    // do something
} else if (condition){
    // do something else
} else {
    // do something else
}
```

TERNARY OPERATOR

syntax

```
(expression) ? trueCode : falseCode;
```

example

```
let age = 12;
let allowed = (age > 18) ? 'yes' : 'no';
// value of allowed is 'no'
```

COMPARISON OPERATORS

`==`	*Equal to — DO NOT USE*
`===`	*Strict equal to*
`!=`	*Not equal to — DO NOT USE*
`!==`	*Strict not equal to*

`>`	*Greater than*
`<`	*Less than*
`>=`	*Greater than or equal to*
`<=`	*Less than or equal to*

LOGICAL OPERATORS

`&&`	*and*
`||`	*or*
`!`	*not*

FALSY VALUES

`false`
`0`
`""`
`NaN`
`null`
`undefined`

FUNCTION DECLARATION

syntax
```
function name(parameters) {
  // do something
}
```

example
```
function speak(name) {
  console.log('Hello, ' + name);
}
```

FUNCTION EXPRESSION

syntax
```
let name = function(parameters) {
  // do something
};
```

example
```
let speak = function(name) {
  console.log('Hello, ' + name);
}
```

ARROW FUNCTION

syntax
```
let name = (parameters) => {
  // do something
}
```

example
```
let speak = (name) => {
  console.log('Hello, ' + name);
}
```

CALLING A FUNCTION

syntax
```
name(arguments);
```

example
```
speak('Michelle');
```

SPECIFYING STRICT MODE

```
'use strict';
     or
"use strict";
```