JavaScript

Making Decisions

Opening Exercise

Download exercise file. Write a function that takes 2 arguments, both are strings: `compareText(selector, text)`
Your function should:

1. Find the first occurrence of the HTML element in the DOM tree that matches the selector that was passed to the function as an argument.
2. Compare its text content to the second argument (use innerText)
3. If the element's content is the same as the second argument, display an alert "we have a match!", if not, displays an alert "no matches!"

Test your function using the console. Try different selectors.

Making decisions

```
if (expression) {
    //execute statements A
}

//execute statements B
```

if **expression** evaluates to true, then:

execute statements **A**,
execute statements **B**.

But if expression evaluates to false, then:

execute statements **B** only.

Making decisions

```
if (x > 0) {
    document.write("greater!");
}

document.write("moving on");
```

Our code:

```
var x = 1;
```

The page in the browser displays:

```
greater
moving on
```

Making decisions

```
if (x > 0) {
    document.write("greater!");
}

document.write("moving on");
```

Our code:

```
var x = -1;
```

The page in the browser displays:

```
moving on
```

Making decisions

```
if (expression1) {
    //execute statements A
}
else if (expression2) {
    //execute statements B
}

//execute statements C
```

if **expression1** evaluates to true:
    execute statements **A**,
    execute statements **C**.
If **expression1** evaluates to false:
    evaluate **expression2**.
    If **expression2** evaluates to true:
        execute statements **B**,
        execute statements **C**.
    But if **expression2** evaluates to false:
        execute statements **C** only.

Making decisions

```
if (x > 0) {
    document.write("greater!");
}
else if (x < 100) {
    document.write("smaller!");
}

document.write("moving on");
```

Our code:

```
var x = 1;
```

The page in the browser displays:

```
greater!
moving on
```

Making decisions

```
if (x > 0) {
    document.write("greater!");
}
else if (x < 100) {
    document.write("smaller!");
}

document.write("moving on");
```

Our code:

```
var x = -101;
```

The page in the browser displays:

```
smaller!
moving on
```

Making decisions

```
if (x > 0) {
    document.write("greater!");
}
else if (x < 100) {
    document.write("smaller!");
}

document.write("moving on");
```

Our code:

```
var x = -5;
```

The page in the browser displays:

```
moving on
```

Making decisions

```
if (expression1) {
    //execute statements A
}
else if (expression2) {
    //execute statements B
}
else {
    //execute statements C
}

//execute statements D
```

if **expression1** evaluates to true:
    execute statements **A**,
    execute statements **D**.
If **expression1** evaluates to false:
    evaluate **expression2**.
    If **expression2** evaluates to true:
        execute statements **B**,
        execute statements **D**.
    If **expression2** evaluates to false:
        execute statements **C**.
        execute statements **D**.

Making decisions

```
if (x > 0) {
    document.write("greater!");
}
else if (x < 100) {
    document.write("smaller!");
}
else {
    document.write("whatever's left!");
}

document.write("moving on");
```

Our code:

```
var x = 1;
```

The page in the browser displays:

```
greater!
moving on
```

Making decisions

```
if (x > 0) {
    document.write("greater!");
}
else if (x < 100) {
    document.write("smaller!");
}
else {
    document.write("whatever's left!");
}

document.write("moving on");
```

Our code:

```
var x = -101;
```

The page in the browser displays:

```
smaller!
moving on
```

Making decisions

```
if (x > 0) {
    document.write("greater!");
}
else if (x < 100) {
    document.write("smaller!");
}
else {
    document.write("whatever's left!");
}

document.write("moving on");
```

Our code:

```
var x = -5;
```

The page in the browser displays:

```
whatever's left!
moving on
```

Testing the condition

The expression we test in an **if** statement must be a **Boolean** expression:
(named after mathematician George Boole

It MUST evaluate to a Boolean value: **TRUE** or **FALSE**

That expression:
- Can be a Boolean value: true or false (*no quotation marks*)
- Can be a variable holding a Boolean value of true or false
- Can be an expression that evaluates to a Boolean value

Testing the condition

An expression will evaluate to a Boolean value when you **compare** two expressions to get a value of true or false:

`a < b` evaluates to true if `a` is less than `b`

We use the following **comparison operators**:

>	greater than	5 > 3	evaluates to true
<	less than	5 < 3	evaluates to false
==	equal	4 == 4	evaluates to true
<=	greater than or equal	etc...	
>=	less than or equal		
!=	not equal		

Testing the condition

An expression will evaluate to a Boolean value when you combine two ***Boolean*** expressions with **logical operators**:

expression1 and expression2 a **&&** b

*evaluates to true if **both** a and b are true*

expression1 or expression2 a **||** b

*evaluates to true if **either** a **or** b are true*

...or when you negate a ***Boolean*** expression with the NOT operator:

not expression1 **!**a

evaluates to true if a is false, and to false if a is true

Using logical operators: truth tables

p	q	not p	p and q	p or q
true	true			
true	false			
false	true			
false	false			

Using logical operators: truth tables

p	q	not p	p and q	p or q
true	true	false	true	true
true	false	false	false	true
false	true	true	false	true
false	false	true	false	false