Lecture 2

Browser Popups, Input from Browser, Switch, Loops

While, for, do-while

Keyword break and continue

IT TALENTS
Training Camp

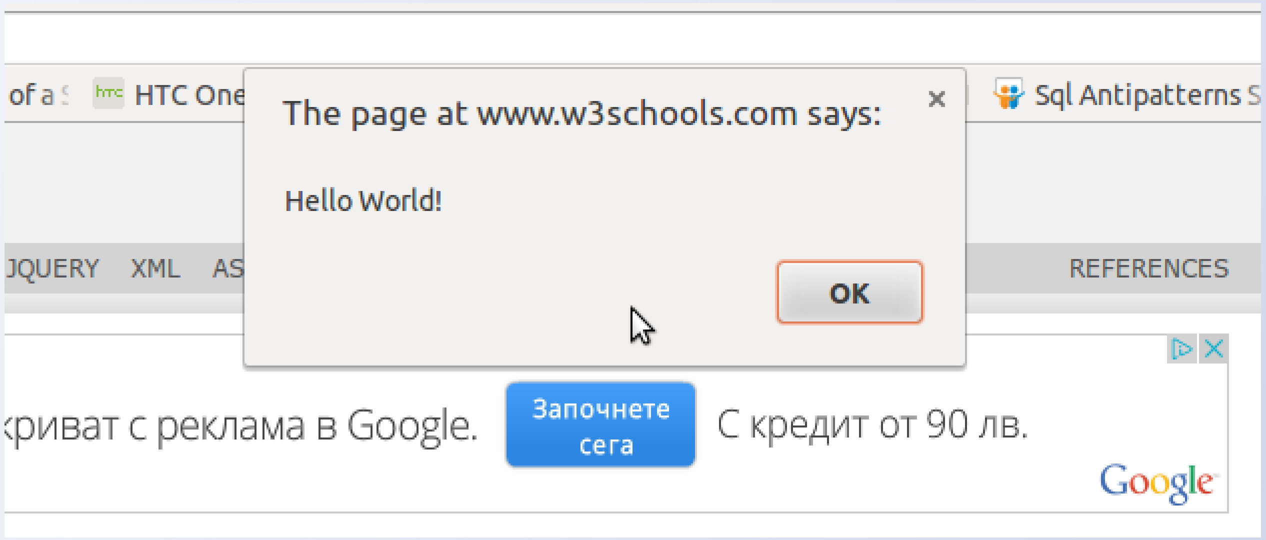
Browser Popups

- alert(<some string>)
- prompt(<some string>)
- confirm(<some string>)

Alert Browser Popup

alert(<some string>) - displays a dialog box in the browser with the string specified and OK button.

Usage – when we want to show something to the user.

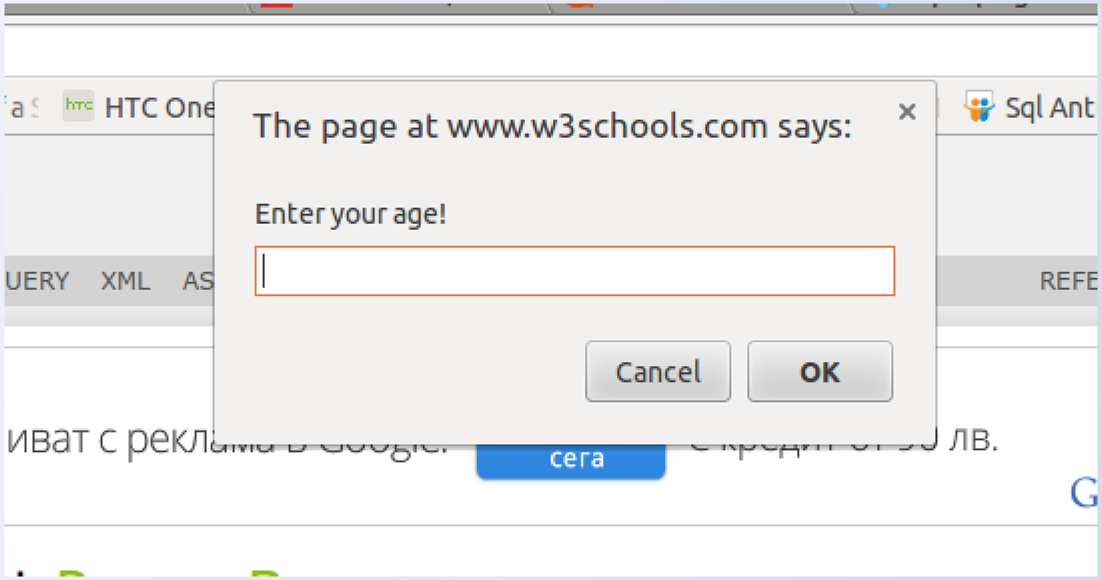

The page at www.w3schools.com says:

Hello World!

IT TALENTS Training Camp

Prompt Browser Popup

prompt(<some string>) - displays a dialog box with the string specified and a text box waiting for input, plus OK and Cancel . Returns null on Cancel and the value of the texbox.



The page at www.w3schools.com says:

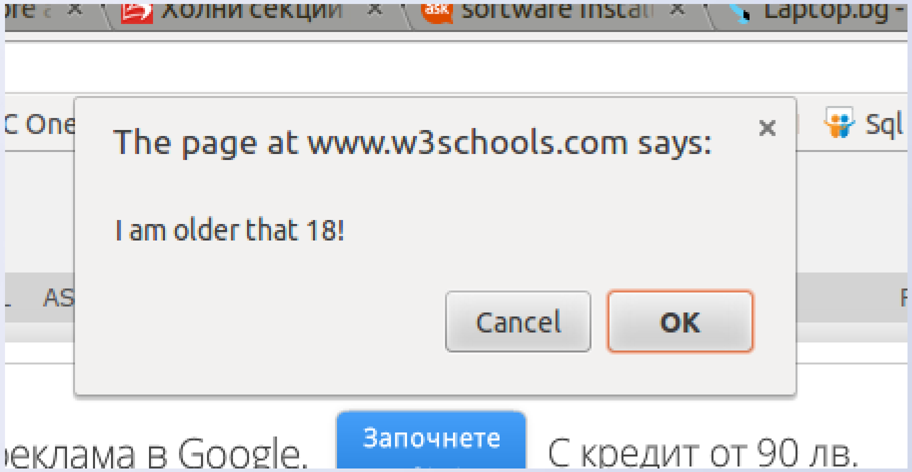
Enter your age!

Confirm Browser Popup

confirm(<some string>) - displays a dialog box with the string specified and OK and Cancel buttons.

Returns ***true*** if the user presses OK and ***false*** if the pressed button is Cancel.

We use it when we want some confirmation from the user.


The page at www.w3schools.com says:

I am older that 18!

Problem

Print in the console by given number the matching day of the week, for example: 1 – monday, 2- tuesday , etc. and 'Error' when there is no matching week day for the given number. The solution with if – else control statement:

```
var day = prompt('Enter a day of the week!');
if (day == 1) {
    alert('Monday');
} else if (day == 2) {
    alert('Tuesday');
} else if (day == 3) {
    alert('Wednesday');
}
//............
else {
    alert('Enter number between 1 and 7');
}
```

Switch Solution

```
var day = prompt('Enter day of the week');
 // switch uses === operator for comparison
day = parseInt(day);
switch (day) {
    case 1:
        alert('Monday');
        break;
    case 2:
        alert('Tuesday');
        break;
    case 3:
        alert('Wednesday');
        break;
    //............
    default:
        alert('Enter number between 1 and 7');
        break;
}
```

Switch Syntax

```
switch(expression) {
    case n:
        code block
        break;
    case n:
        code block
        break;
    default:
        default code block
}
```

Problem

Print all the numbers

- From 1 to 5
- From 1 to 1000
- From 1 to n
- From n to m

What is a loop?

- A loop is a structure that allows sequence of statement to be executed more times in a row

- Loops have a boolean condition and a block of code for execution. While the condition is true, the block is being executed.

- A loop that never ends is called an infinite loop

While

- While the condition is true, the block is being executed.

While

Counter initialization

Boolean condition.
If i > 100, the next block will be skipped

```
var i = 1;

while (i <= 1000) {
    console.log(i);
    i++;
}
```

Block of code
repeatable execution

do-while

- Similar to while loop but always enters the execution at least once because Condition is after the execution

Execute the block of code

```
do {
    console.log(i);
    i++;
} while (i <= 1000)
```

Check if i<=1000. If it's true, repeat once more.

For loop

- Consists of
  - Initialization
  - condition
  - Update statement
  - body

If i becomes equal or
bigger than the length of the array,
the loop will quit.

Initialization

Update statemen

```
for (var i = 0; i < 10; i++) {
    console.log(i);
}
```

Condition

Body

IT TALENTS
Training Camp

Problem

- Try to quit a for-loop during the execution of the repeatable block
-
- One possible to solution is to set the counter to a value which will make the boolean condition quit the loop....but...

Break

- Break is a keyword
- A statement by itself
- It doesn't require anything else
- It stops the execution of the loop

The loop will quit
when i is equal to 7

```
for (var i = 0; i < 50; i++) {
    if (i == 7) {
        Break;
    }
}
```

Problem

- Try to omit specific block of code in the body – for example sum all numbers between 1 and 100 but omit all numbers between 51 and 74
-
- Encapsulating the code in if-else statements may be used. Although for more complicated structures should be used for more complicated cases

Continue

- Continue is a keyword
- A statement by itself
- It doesn't require anything else
- It stops the current iteration of the loop, but doesn't stop the loop

```
for (var i = 0; i < args.length; i++) {
    if (i > 51 && i < 71) {
        continue;
    }
    sum = sum + i;
}
```

If i is between 51 na 74,
the loop will skip
all statements after **continue**.

Summary

- Browser popups
- Switch statement
- Why do we use loops?
- What does a loop consist of?
- Difference between *while* and *do-while*?
- How to use *for* – loop?
- How to terminate a loop?
- How to stop the current iteration?

IT TALENTS
Training Camp