

A decorative graphic on the right side of the slide. It features a large, semi-transparent blue circle. Inside this circle is a smaller, semi-transparent green circle. Within the green circle is a pink circle containing a white, stylized letter 'G'. The background of the pink circle is filled with white binary code (0s and 1s).

JavaScript Lecture 2

Code Girls 2020-21



onclick

- Executes JavaScript code when a button is clicked
- Write the code that you want to execute in a function. In this case, myFunction() would be executed

```
<button onclick="myFunction()">Click me</button>
```



alert()

- Displays an alert box with an OK button

```
<button onclick="myFunction()">Display Alert</button>  
<script>  
function myFunction() {  
  alert("Hello\nHow are you?");  
}  
</script>
```





Comparison Operators

- Comparison operators are used to determine equality or differences between variables of values
- Equal to: `==`
- Equal value and equal type: `===`
- Not equal: `!=`
- Not equal value or not equal type: `!==`
- Greater than: `>`
- Less than: `<`
- Greater than or equal to: `>=`
- Less than or equal to: `<=`
- When comparing a string and a number, JavaScript converts the string to a number. If the string is empty, it is converted to 0. If it is a non numeric string, it is converted to NaN which is false
- When comparing two strings, they are compared alphabetically. So, “2” is greater than “12” because 1 comes before 2.



Logical Operators

- And: `&&`
 - `(x < 10 && y > 1)`
- Or: `||`
 - `(x == 5 || y == 5)`
- Not: `!`
 - `!(x == y)`



Conditions

- Conditional statements are used for when you want to perform different actions for different decisions
- **if**: Executes the block of code if the condition is true
- **else**: Executes the block of code if the if statement is false
- **else if**: Executes the block of code if the previous conditions are false and this condition is true

```
if(time < 10){
    alert("You won!")
}
else if(time >= 10 && time <= 100){
    alert("Try again next time")
}
else{
    alert("Please enter a number from 1-100")
}
```



For Loops

- Used when you want to run the same code several times, each time with a different variable

```
for (i = 0; i < 5; i++) {  
    text += i + "<br>";  
}
```



While Loops

- The block of code is executed for as long as the condition is true

```
var text = "";  
var i = 0;  
  
while(i<5){  
    text += i + "<br>";  
    i++;  
}
```

- Be careful of infinite loops!



Break and Continue

- The break statement “jumps out” of a loop and continues to execute the code after the loop
- The continue statement “jumps over” one iteration in the loop

```
//jumps out of the loop of the value is 3
```

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { break; }  
    text += "The number is " + i + "<br>";  
}
```

```
//skips over the value 3
```

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { continue; }  
    text += "The number is " + i + "<br>";  
}
```



Form Validation

- We can use JavaScript to check if a form is filled out correctly

```
function validateForm() {  
    var x = document.forms["myForm"]["fname"].value;  
    if (x == "") {  
        alert("Name must be filled out");  
        return false;  
    }  
}
```

```
<form name="myForm" action="/action_page.php" onsubmit="return validateForm()"  
method="post">  
Name: <input type="text" name="fname">  
<input type="submit" value="Submit">  
</form>
```



Acknowledgments

Thanks to:

- Ms. Lola Piper for sponsoring our club and guidance