JavaScript 3

Recitation 8 for CS170
Function Syntax

- A function is **a block of code** that will be executed when "someone" calls it:

- The "function" keyword followed by **function name**, **argument list (can be void)** and **curly braces** containing code

Example:

```javascript
function greeter(student_name , course_num) {
    document.write("Welcome" + student_name + "to" + course_num);
}
```
Function names

- They begin with a letter, use any mix of letters, numbers, and underscores (_), **avoid reserved words** (function, var, if, else).

- Also give them meaningful names.
Functions With a Return Value

- Sometimes you want your function to return a value back to where the call was made.
- This is possible by using the `return` statement.
- Example:

```javascript
function myFunction() {
    var x=5;
    return x;
}
```
Calling JavaScript Functions

- The code inside the function will be executed when the function is called. The function can be called directly when an event occurs like when a user clicks a button, and it can be called internally by JavaScript code too.

- Example 1:

```html
<button onclick="greeter('Student','CS170')">Greetings</button>
```

- Example 2:

```javascript
var name, course;
if(user == student) { greeter(name, course); }
```
## Scope of variables

- The scope of a name defines how “far” from its declaration it can be used.

<table>
<thead>
<tr>
<th></th>
<th>Local Variables</th>
<th>Global variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declare position</td>
<td>Inside a function</td>
<td>Outside a function</td>
</tr>
<tr>
<td>Range of use</td>
<td>Only Inside of this function(code block)</td>
<td>Throughout the whole program</td>
</tr>
<tr>
<td>Life time</td>
<td>deleted when the function is completed.</td>
<td>deleted when you close the page</td>
</tr>
</tbody>
</table>
JavaScript Built-in Functions

- We have learnt:
  - Alert box, prompt box, confirm box.

- More useful build-in functions:
  - Date()

The Date object is used to work with dates and times.

Date().toString():

Converts a Date object to a string

- Math.max(n1,n2,n3,...,nX)

- `<script type="text/javascript"> var str = new String("Hello world"); alert(str.fontcolor("red" )); </script>`
Date()

- var myDate = new Date(year, month, day, hours, minutes, seconds, milliseconds);
- var myDate = new Date();
- Date().

<table>
<thead>
<tr>
<th>getDate()</th>
<th>getDay()</th>
</tr>
</thead>
<tbody>
<tr>
<td>getFullYear()</td>
<td>getHours()</td>
</tr>
<tr>
<td>getMilliseconds()</td>
<td>getMinutes()</td>
</tr>
<tr>
<td>getMonth()</td>
<td>getSeconds()</td>
</tr>
<tr>
<td>getTime()</td>
<td>getTimezoneOffset()</td>
</tr>
</tbody>
</table>
String()

- A JavaScript string simply stores a series of characters.

- A string can be any text inside quotes. You can use single or double quotes.

  Eg. var carname = "Volvo XC60";

- You can use quotes inside a string, as long as they don’t match the quotes surrounding the string.

  Eg. var answer = "He is called 'Johnny'";
you can put quotes inside a string by using the `\` escape character:

```javascript
var answer = "He is called \"Johnny\""
```

**String Length:** The length of a string (a string object) is found in the built in property `length`:

```javascript
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = txt.length;
```
String()

- `indexOf()`: Returns the position of the first found occurrence of a specified value in a string

- Eg

  ```javascript
  var str = "Please locate where 'locate' occurs!";
  var pos = str.indexOf("locate"); //pos = 7
  ```

- The `lastIndexOf()` method returns the index of the last occurrence of a specified text in a string:

  ```javascript
  var pos = str.lastIndexOf("locate"); //pos = 21
  ```
# String()

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>charAt()</td>
<td>Returns the character at the specified index (position)</td>
</tr>
<tr>
<td>charCodeAt()</td>
<td>Returns the Unicode of the character at the specified index</td>
</tr>
<tr>
<td>concat()</td>
<td>Joins two or more strings, and returns a copy of the joined strings</td>
</tr>
<tr>
<td>fromCharCode()</td>
<td>Converts Unicode values to characters</td>
</tr>
<tr>
<td>indexOf()</td>
<td>Returns the position of the first found occurrence of a specified value in a string</td>
</tr>
<tr>
<td>lastIndexOf()</td>
<td>Returns the position of the last found occurrence of a specified value in a string</td>
</tr>
<tr>
<td>localeCompare()</td>
<td>Compares two strings in the current locale</td>
</tr>
<tr>
<td>match()</td>
<td>Searches a string for a match against a regular expression, and returns the matches</td>
</tr>
<tr>
<td>replace()</td>
<td>Searches a string for a value and returns a new string with the value replaced</td>
</tr>
<tr>
<td>search()</td>
<td>Searches a string for a value and returns the position of the match</td>
</tr>
<tr>
<td>slice()</td>
<td>Extracts a part of a string and returns a new string</td>
</tr>
<tr>
<td>split()</td>
<td>Splits a string into an array of substrings</td>
</tr>
<tr>
<td>substr()</td>
<td>Extracts a part of a string from a start position through a number of characters</td>
</tr>
<tr>
<td>substring()</td>
<td>Extracts a part of a string between two specified positions</td>
</tr>
</tbody>
</table>
JavaScript in class practice

- Design an auction system.
- Use 6 prompt boxes ask for three users names and their bids and store them in some variables.
- Define a function to determine the winner (the largest bidder) using `Math.max()`.
- Create a button, when clicking on it, print the name of the winner. (use `document.write()`)
- Hint: To determine which name to print out, you may use `if` statements.