The JavaScript Language

An introduction

What can you do with JavaScript

- A lot:
  - Process (limited) user input, get data from external source (databases and more), do any calculations, generate HTML (from previous slides)
- Most importantly, for our class:
  - Access and modify a web page (both content and HTML markup)
  - Do that as the user is viewing the page

1. ACCESS CONTENT

Select text, elements, or attributes on an HTML page

1. ACCESS CONTENT

Select text, elements, or attributes on an HTML page

2. MODIFY CONTENT

Add text, elements, or attributes to an HTML page (or remove them)

1. ACCESS CONTENT

Select text, elements, or attributes on an HTML page

2. MODIFY CONTENT

Add text, elements, or attributes to an HTML page (or remove them)

3. PROGRAM RULES

Specify a set of steps for the browser to follow (like a recipe)

1. ACCESS CONTENT

Select text, elements, or attributes on an HTML page

2. MODIFY CONTENT

Add text, elements, or attributes to an HTML page (or remove them)

3. PROGRAM RULES

Specify a set of steps for the browser to follow (like a recipe)

4. REACT TO EVENTS

Tell a script to run in response to some event that occurred

What you will need to learn

1. Programming concepts:
    - How to organize your ideas about sequences of tasks)

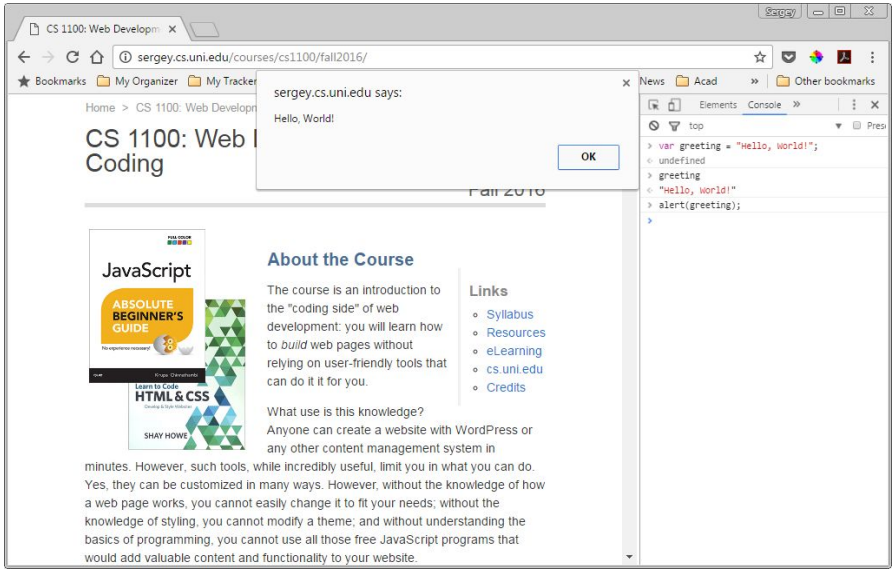
2. The JavaScript language:
    - How to express these ideas using the language vocabulary, syntax, and grammar

* You will also benefit greatly from looking at how this is done:

- Look at examples of code
- Study tutorials
- Use stackoverflow.com

Where to write your JavaScript code

- In your favorite text editor - just like you did with HTML and CSS
- You may use the browser console for trying out code



Where to place your JavaScript code

a. Inside the browser:
   - inside <script> </script> tags
b. In a separate file (preferred approach):
   - use any name with a ".js" extension
   - do NOT include the <script> tags in your .js file
   - link to your HTML using this tag: <script src="path-to-your-.js-file">
c. Both (only if you really need to)

* Make sure to place your <script> tag close to the bottom of your HTML file

JavaScript statements

- Each “step” in a program is a statement
- A statement is code that performs a task
- Statements are terminated with a semicolon:
  - var message;
    message = “Hello, World!”;
  - var x; x = 42;
  - **recommended style: one line = one statement (or partial statement)**
- There are different types of statements; today we take a closer look at assignment statements

Variables

1. Declaring a variable
2. Naming a variable
3. Assigning a value to a variable (assignment statement)

```
var quantity;
```

```
var quantity;
```

KEYWORD

```
var quantity;
```

VARIABLE NAME

Naming variables

- JavaScript is case-sensitive: message <> Message <> MESSAGE
- Naming variables:
  - start with a character, “$”, or underscore
  - do not use spaces
  - do not use keywords (var, for, if, else, function, etc...)
  - use camelCasing when name consists of more than one word
  - use descriptive names,
    - that are short

```
quantity = 3;
```


quantity = 3;
VARIABLE NAME

```
quantity = 3;
```

ASSIGNMENT OPERATOR

```
quantity = 3;
```

VALUE

Assignment statement

- Declaring and assigning variables:
  - var message;
    message = “hello!”;
  - var message = “hello!”;
- Assignment statement:
  - [variable] = [expression]
    - a = 1;
    - a = 1 + 2;
    - b = 2;
    - a = b;
    - a = a + 1;
- Assignment operator IS NOT the same as equality operator:
  - a = b <> a == b <> a === b (more on this later!)

The browser as context

- Use any simple web page
- Use the console in your browser to write your code
- use document.querySelector(“**your-selector**”) to access elements in your HTML
- assign these elements to variables
- use **your-variable**.innerHTML = “new value” to modify each variable