CSCI 341 : HANDS ON

UNIT: Conversions

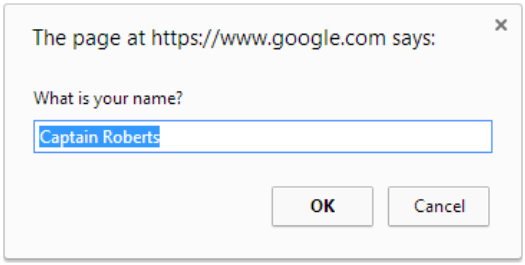
Activities

- Explore **variables, prompts, and concatenation**.
- Explore **JavaScript's weak typing** and **data type conversion**.
- Practice with variables, prompts, alerts, data type conversion and concatenation.

Activity One

The "prompt" method displays a dialogue box which asks the user for input. When calling the prompt method, you can send it either one or two strings as arguments. The first string contains the text displayed by the dialogue box, often asking the user a question. The second string is optional, but if used will enter a default value for the input. The following is an example of prompt in use.

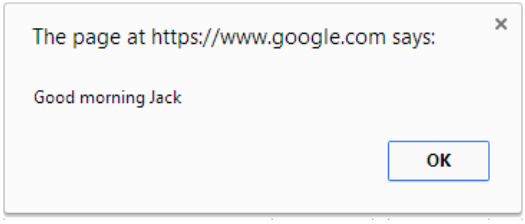
```
var strName = "";
var strQuestion = "What is your name?";
var strDefault = "Captain Roberts";
strName = prompt(strQuestion, strDefault);
```



Activity Two

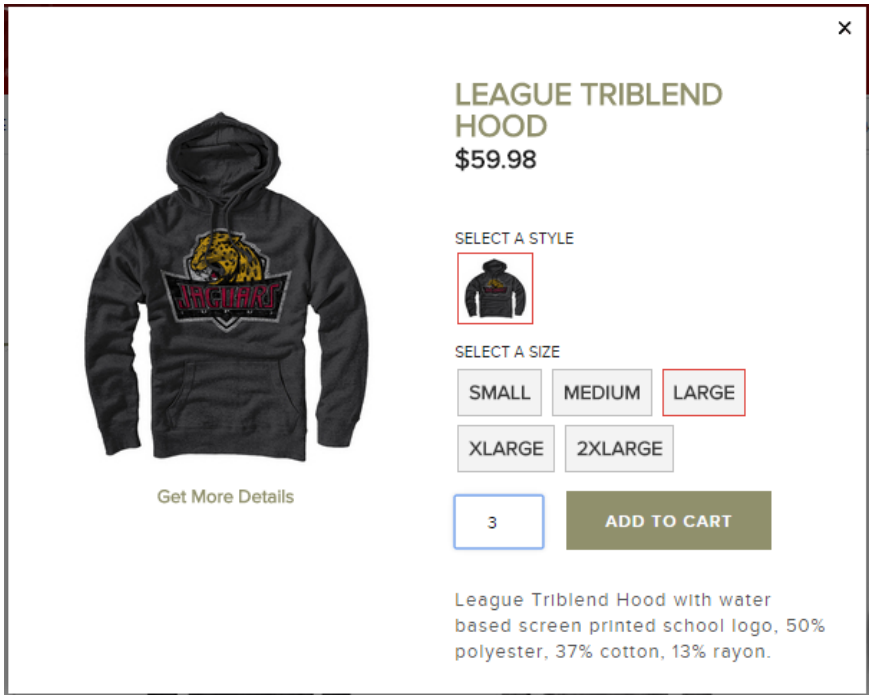
Concatenation is just a fancy word for adding two or more strings together. To add two or more string variables together, use the + operator. The following is an example of concatenation.

```
var strOutput = "";
var strName = "Jack";
var strGreeting = "Good morning ";
strOutput = strGreeting + strName;
alert(strOutput);
```



Activity Three

In web work, much of our user interactivity comes from user-inputted data. For example, a user fills out a text field in a form, and Javascript is used to read and process this information. Or a user enters quantity data in a shopping cart app.

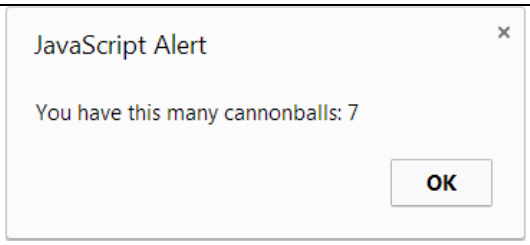


It is important to realize that the data the user enters "comes in" as string data.

```
var strNumber = "7";    // Original number of cannon balls
var strNewNumber = "";  // Calculated new number of cannon balls

var intNumber;          // strNumber stored as an integer
var intNewNumber;       // strNewNumber stored as an integer

alert("You have this many cannonballs: " + strNumber);
```



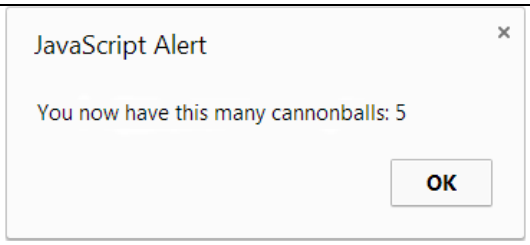
If your application needs to perform numeric calculations with the data, you'll need to convert the string data to numeric form. This is such a common requirement that the language provides built-in utilities for data conversion.

The two main work horses on data conversion from strings to numerics are parseInt() and parseFloat().

parseInt() takes a string and converts it to an integer; parseFloat() takes a string and converts it to a decimal (floating point) number. The following is an example of parseInt.

```
intNumber = parseInt(strNumber);          // Convert strNumber to integer for calculations
intNewNumber = intNumber - 2;             // Throw 2 cannonballs overboard. Plunk! Plunk!
strNewNumber = intNewNumber.toString(); // Convert intNewNumber to a string

alert("You now have this many cannonballs: " + strNewNumber);
```



Activity Four

(Note: all modern languages support a kind of variable called a constant. I know, that's sort of oxymoron-ish: a constant variable. But! this construct provides some defensive programming.) Let's say you are running a shopping cart app and you are calculating sales tax. Sales tax does occasionally change, but shouldn't as part of normal program calculations. As a protected sanity check, you might set up sales tax as a constant, so that your program wouldn't write to that variable location by mistake. Sadly, Javascript constants have not been uniformly supported by the browsers. At this time, "const" has basic support on Internet Explorer, Firefox, Chrome, Safari, and Opera, but Internet Explorer did not start supporting it until IE 11. So, know that constants exist, but right now are not a good tool to utilize since you won't know what browser is being used to interpret your Javascript code. Below is an example of a JavaScript constant.

```
const price = 287;
```

Lab Instructions

1. Create an HTML5 page and JavaScript file that prompts the user for their pirate name. Use a version of the prompt that includes a default pirate name. For example: "Peg Leg O'Brien"
2. Store the result in a variable with the "str" prefix.
3. Prompt the user for the number of gold doubloons they plundered - also with a default value.
4. Store the number id a variable with the "str" prefix.
5. Perform parseInt() on the number of gold doubloons to convert the string to an integer.
6. Multiply the number of gold doubloons by the worth of a gold doubloon in USD ($287) to determine the profit.
7. Store the calculated profit in a variable.
8. Output concatenated string including user's pirate name, the number of gold doubloons, and total profit in USD using textContent. For example, "I, Peg Leg O'Brien, have plundered 17 gold doubloons for a total profit of $4879. Argh!
9. After validating all files, be sure to upload it to the server.
10. Follow all procedures listed below.

Important Procedures for All Labs

Here are some general notes for perfection that you should follow for every assignment:

1. Please produce all web content to HTML5 standards.
2. Please **validate** all your files.
3. Be sure to update the header block comments for each file.
4. Be sure to check your browser's console / developer tools for error free code.
5. Test your code in Chrome and Edge at a minimum.
6. Use only your own original code for all labs.
7. Be sure to put your CSS and JavaScript in a separate files from your html.
8. Be sure to read through the lab rubric in Canvas.
9. Submit your lab in Canvas for grading.

Holler if you have any questions!

Mission Accomplished!

Congrats, you have completed this week's assignments! Here is your reward: How much do you pay a pirate for an ear piercing? … A buccaneer! Arg, that be a bad joke.