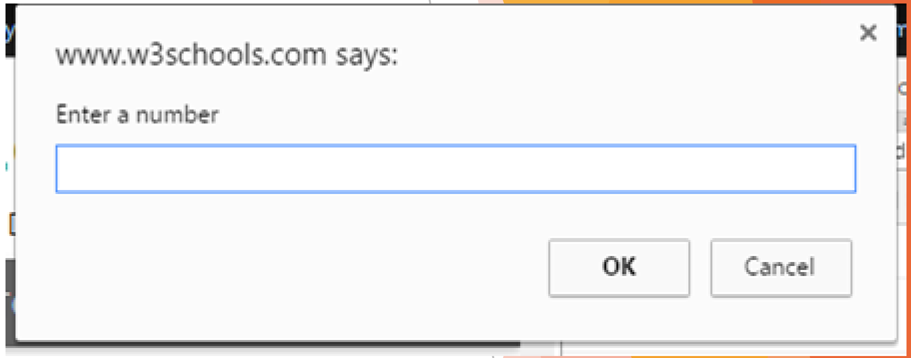
Conditional Execution, I/O

Getting User Input

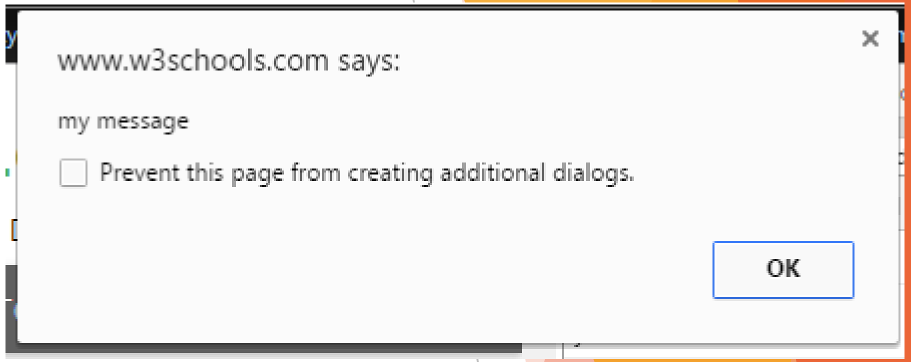
- prompt()
  - Command used to get user input from a dialog box
  - When we write
    - `let userInput = prompt('Enter a number');`



  - A dialog box will appear, and the JavaScript program will wait until the user enters an input before proceeding
  - This is called a **modal** dialog box
  - In the above code, what the user typed in the dialog box is stored in the variable **userInput** as a **string**

Displaying Output

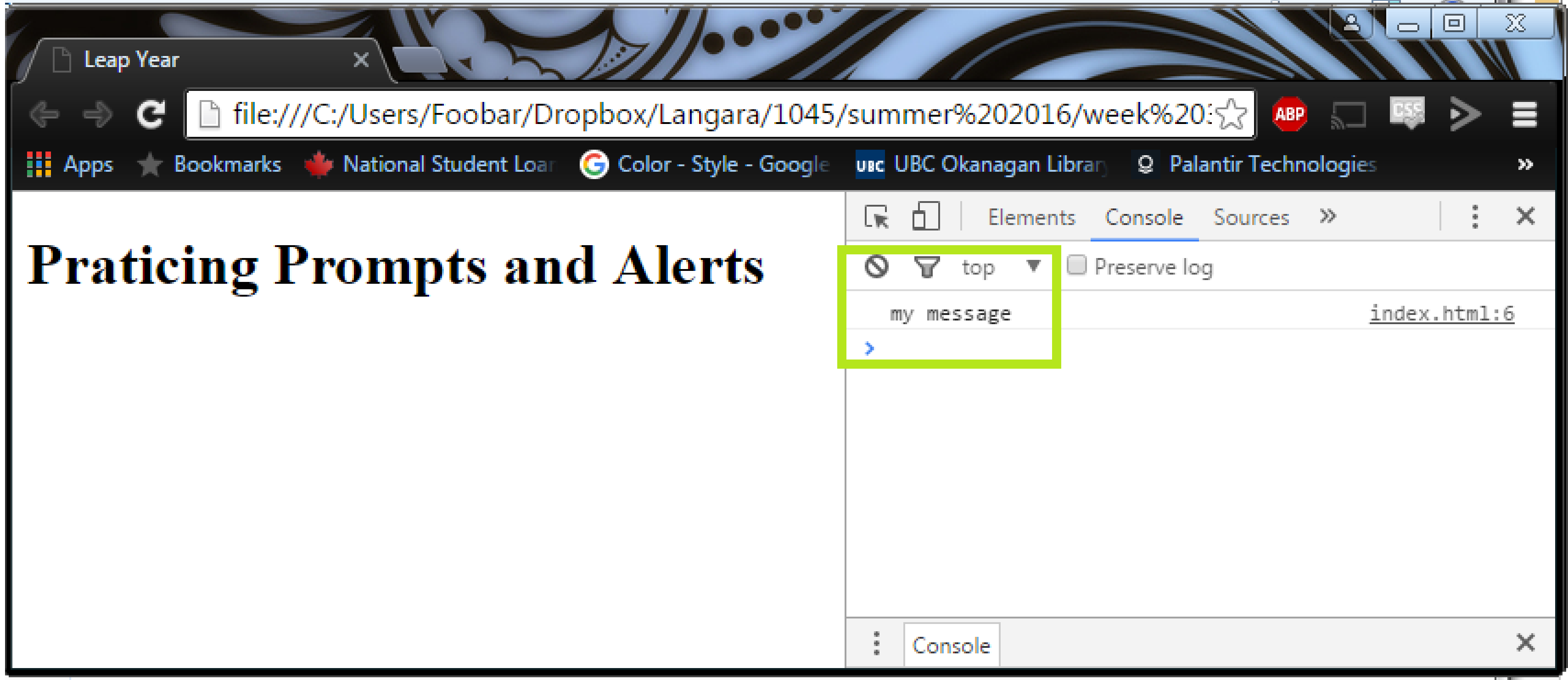
- Alert() displays a message to the user a message through a dialog box.
  - Ex. `alert("my message");`
- You can also write to the browser console
  - Ex. `console.log("my message");`



www.w3schools.com says:

my message

Prevent this page from creating additional dialogs.



Praticing Prompts and Alerts

```
my message                                   index.html:6
>
```

Making Decisions

- We can use JavaScript to make decisions
  - Uses Boolean Expressions
  - If you don’t have a Boolean expression, you can use comparisons to create one
- All data that we use for making decisions must be in a comparable format
  - Strings
  - Booleans
  - Numbers
- Because these are the only things JavaScript knows how to compare

Decisions Computers Can’t Make

- If we can’t quantify, “stringify”, or “booleanify” our inputs/data, then a computer can’t base a decision on that data
  - For Example: Should I eat Cake or Ice Cream?
- The above is a true or false question, but how can a computer make a meaningful decision
- If the data does not contain enough information to answer the question!
- The computer can’t quantify all of the factors that will affect my decision
  - Google: Cannot tell the age of a person from a picture
  - We cannot predict earthquakes far in advance, because we haven’t found meaningful data(statistically significant) that can predict it

Boolean Expressions

- Comparison operators ===, >=, <= will result in a boolean expression
  - Ex. `num >= 5;`
- Logical operators, && and || are how we combine Boolean expressions.
  - Ex. `num >= 5 && num <= 9;`

```
let num = 10;
let result = num >= 5 && num <= 9;
```

- So what is stored in result after these two statements execute?

If Statements

- If statements are used for conditional execution
- Up till now, we have run our JavaScript top to bottom without skipping a line
- If statements allow us to skip lines of code
- Or to only execute certain lines of code IF some condition has been met

```
if(<boolean condition>) {
     //Only executed when true
}
```

If Else Statements

- We can add an else statement
- The else code block will only execute when the if condition evaluates to FALSE

```
if(<boolean condition>) {
   //Only executed when condition is true
} else {
   //only executed when condition is false
}
```

Example Using If Else

```
let greeting = “”;
let hour = prompt(“Enter the hour in 24hr format”);
console.log(typeof(hour));
hour = parseInt(hour);

if (hour < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

Note

- Do not forget that *MOST* lines of JavaScript end in a semi colon

  Ex.

  ```
  console.log(“Hello World!”);
  ```

- One of the exceptions to this rule is when we are writing if statements

```
if (hour < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

Code Blocks

- If statements naturally only execute the FIRST line following the if statement

```
if(height > 180)
    alert("you are above average height");
```

- This is where code blocks come in handy
- A code block is a group of JavaScript statements surrounded by curly braces { }
- If you follow an if statement with a code block, then all statements in the { } will execute

```
if(height > 180){
    alert("you are tall.");
    alert("your height in inches is 70.86" );
}
```

Exercise

- What’s wrong with the following code fragment? Rewrite it so that it produces the correct output. Assume total, max, and sum have already been defined.

```
if(total === max)
    if(total < sum)
        console.log(“The total equals max and is less than sum”);
else console.log(“The total is not equal to max”);
```

Question

- What output is produced by the following code?

```
let num = 87, max = 25;
if (num >= max*2)
   console.log(“apple”);
   console.log(“orange”);
console.log(“banana”);
```

If, Else If, Else

- You can add an else if statement to the traditional if else structure
- This is used when you have more than two possible cases to execute

```
if(<condition 1>) {
   //executed if condition 1 is true
} else if(<condition 2>) {
  // executed if condition 1 is false, and
condition 2 is true
} else {
  //executed when all conditions are false.
}
```

Example Using Else If

```
let greeting = “”;
let time = parseInt(prompt(“Enter the hour in 24hr format”));

if (time < 10) {
    greeting = "Good morning";
} else if (time < 20) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

Question

- What’s wrong with the following code?

```
if(length = 100)
   console.log(“The length is 1 meter”);
```

Practice Writing It

- At the end of a game, a player’s score determines what message is printed to the screen
    - If the user’s score is 100, the message “Great Success!” is displayed
    - If the user’s score is under 100 but above 50, the message “Good Work!” is displayed
    - If the user’s score is equal to or under 50, the message “Better luck next time!” is displayed
- How do we write this information using if statements?

```
let score = parseInt(prompt(“Enter your score!”));
if(score === 100)
   alert(“Great Success”);
else if (score > 50)
   alert(“Good Work!”);
else alert(“Better luck next time!”);
```

Nesting If Statements

- For more complex conditions you can nest an if statement inside of an if statement
- So for example, if you wanted to see if someone was both male and tall you could write

```
if(gender === “male”){
   if(height > 190)
      alert(“you are a tall male”);
   else alert(“you are a short male”);
}else{
   if(height > 190)
      alert(“you are a tall female”);
   else alert(“you are a short female”);
}
```

Alternative to Nesting If Statements

```
if(gender === “male” && height > 190)
    alert(“you are a tall male”);
else if(gender === “male” && height < 190)
    alert(“you are a short male”);
else alert(“you are female”);
```

a	b	Result of a || b
T	T	T
T	F	T
F	T	T
F	F	F

Example Using Logical Or

- Remember the logical or

```
let input = prompt("Enter your favorite fruit");
if (input === "strawberry" || input === "pineapple")
   alert("That's my favorite fruit too!");
else alert(“Not the best fruit");
```

Question

- What output is produced by the following code?

```
let limit = 100, num1 = 15, num2 = 40;
if (limit <= limit){
  if (num1 === num2)
    console.log("lemon");
  console.log("lime");
}
console.log("grape");
```

Verifying Input

- When we obtain input form the user we have to make sure it is what we expect
- We can use if statements to verify that the user has entered valid input
- We can check if the something is a number using the isNaN() function
  - `Ex. isNaN(100) returns false`
  - `Ex. isNaN("100") returns false`
  - `Ex. isNaN("MOM") returns true`
  - `Ex. isNaN(NaN) returns true`

Example Verifying User Input

```
let num1 = prompt("Please enter a number");
if(isNaN(num1))
   alert("That wasn't a number! Idiot!");
else alert("Thank you for complying.");
```

Review: Use If to Make Decisions

- Once we have obtained valid input from the user, we have to decide what to do with it
- #teamIronMan   #teamCap   #civilWar #neutral

```
let team = prompt("Enter cap or iron to pick your team!");
if(team ==="cap")
   alert("Avengers Assemble!");
else if(team === "iron")
   alert("Genius. Billionaire. Playboy. Philanthropist.");
else alert("Pick a side.");
```

Switch Statements

- Are like a condensed version of several if, else if, else statements

```
switch(<expression>)
{
     case <expression1>:
         //code block 1
         break;
     case <expression 2>:
        //code block 2
         break;
      default:
        //default code block
         break;
}
```

How Switch Works

- Switch compares the expression in the switch statement, with each of the case expressions using the === operator
- The comparison starts from the top case and continues until it finds one that is true
  - If a true case is not found then the default case will be executed
- However, if no default is specified, and no true cases are found, then no code block is executed.
- The break statements are **necessary**, because switch statements have fall through behavior (i.e., without the break; one case after another would execute)

Exercise

Rewrite the following switch statement using if-else statements ONLY

```
let grade = prompt ("Enter a numeric grade (0 to 100): ");
let category = Math.round(grade / 10);
console.log("That grade is ");
switch (category){
    case 10:
        console.log ("a perfect score. Well done.");
        break;
    case 9:
        console.log ("well above average. Excellent.");
        break;
    case 8:
        console.log ("above average. Nice job.");
        break;
    case 7:
        console.log ("average.");
        break;
    case 6:
        console.log ("below average. You should see the instructor.");
        break;
    default:
        console.log ("not passing.");

}
```

Solution

```
if(category === 10)
   console.log("a perfect score. Well Done.");
else if (category === 9)
   console.log("well above average. Excellent.");
else if (category === 8)
   console.log("above average. Nice job.");
else if (category === 7)
   console.log("average.");
else if (category === 6)
   console.log("below average. You should see the instructor.");
else console.log("not passing.");
```

Now You Try It

- An insurance company has different insurance base rates for different types of cars. The company rate chart is below, implement the switch statement that prints the correct rate based on the vehicle type.

Car Type	Rate
Compact	$800
Luxury	$1200
SUV	$1100
Van	$1000
Sedan	$1150
Other	$1200

SOLUTION

```
let carType = prompt("Enter the type of car: ");
let rate = 0;
switch(carType)
{
     case 'Compact':
          rate = 800;
          break;
     case 'Luxury':
         rate = 1200;
          break;
     case 'SUV':
         rate = 1100;
          break;
     case 'Van':
         rate = 1000;
          break;
     case "Sedan":
         rate = 1150;
          break;
     case 'Other':
         rate = 1200;
          break;
}
alert("Your insurance rate is: " + rate);
```

Operators Revisited

- The Unary Operators
  - Effect ONE expression, not two like binary operators do
- !
  - negation
- + and -
  - Positive (or numerical type conversion) and negative
- ++ and --
  - Increment and decrement
  - x++ means x = x + 1;

```
let bool = 15 + 2 > 20;
```

!bool will be true

```
let user = +prompt("enter your favorite number.");
user = -user;
```

user will be 25
then -25
as NUMBERS

```
user++;
user--;
```

user will be -24
then -25

Operators Revisited

- ?:
  - The ternary operator
  - Effects THREE expressions
  - Basically a short handed if else statement

```
(<CONDITION>)? //DO IF TRUE : //DO IF FALSE;

let user = prompt(“Enter your gender: “);
(user === “male”)? alert(“you are male”) : alert(“you are female”);
OR
alert((user === “male”)? “you are male” : “you are female”);
```

Exercise

- Write a statement using the ternary operator that checks if a variable called age is less than 30.
- If the age is less than 30 print the statement “You are young!”, otherwise print the statement “You are old!”.

The Remainder Operator

- %
- X % Y will return the remainder after X has been divided by Y

```
Ex.
10 % 7 returns 3
100 % 2 returns 0
```

- We can use remainder to check if a number is even or odd

```
let num = parseInt(prompt(“Enter a number”));
if(num % 2 == 0)
   alert(‘Your number is even.’);
else alert(“Your number is odd.”);
```

Doing Math in JS

- Math Object
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math
- Some great math methods in this class
- `Math.pow(5,2) returns` 5^2
- `Math.round(2.66) returns 3 while Math.round(2.33) returns 2`
- `Math.ceil(5.3) returns 6`
- `Math.floor(10.7) returns 10`
- `Math.min(2,4,6) returns 2`
- `Math.max(2,4,6) returns 6`
- `Math.random() returns a random number between 0.0 and 1.0`

Examples Using Math.random()

```
let random1 = Math.random();
let random2 = Math.random();

let integer = Math.round(random1);
if(integer === 0)
  console.log("Your coin landed on heads!");
else console.log("Your coin landed on tails!");

console.log(Math.min(random1,random2));
```

Generating Specific Random Numbers

- Math.random() returns a random number between 0.0 and 1.0 (not including 1.0)
- We can use this to generate random numbers that fall between specific digits
- For example, if we want to generate a random **integer** between 0 and 100

```
let random = Math.random() * 100;
random = Math.round(random);
```

- For example, if we want to generate a random **integer** between 1 and 100

```
let random2 = Math.random() * 99;
random2 = random2 + 1;
random2 = Math.round(random2);
```

Manipulating Strings in JS

- String Object
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String
- `let str = “hello my name is Inigo Montoya”;`
- `str.length;`
  - will return the length of the string (how many char, including spaces)
- `str.indexOf(‘a’);`
  - will return the position of the first ‘a’ in the str
- `str.lastIndexOf(‘a’)`
  - returns the position of the last ‘a’

Manipulating Strings in JS

- `str.substring(6,16);`
  - returns “my name is”
- `str.substring(17);`
  - returns “Inigo Montoya”
- `let newStr = str.replace(‘Inigo’, ‘Inconceivable’)`
  - newStr will now contain the string “hello my name is Inconceivable Montoya”

Manipulating Strings in JS

- `let str = 'Inigo Montoya';`
- We can convert a string to uppercase and lowercase
- `let bigStr = str.toUpperCase();`
  - bigStr will be 'INIGO MONTOYA'
- `let lilStr = str.toLowerCase();`
  - lilStr will be 'inigo montoya'
- We can get a single character out of string
- `str.charAt(0);`
  - will return I

Manipulating Strings in JS

- `let str = 'Inigo Montoya';`
- `let bigStr = str.toUpperCase();`
  - bigStr will be 'INIGO MONTOYA'
- `let lilStr = str.toLowerCase();`
  - lilStr will be 'inigo montoya'
- `str.charAt(0);`
  - will return I

```
Ex.
let name = "";
name += bigStr.charAt(0);
name += bigStr.charAt(6);
name += lilStr.substring(7,11);

Now what is stored in name?
```

Exercise

- Write the JavaScript that prompts for the user to enter their first name and last name (separately).
- Then print to the console a string composed of the first letter of the user’s first name, followed by the first five characters of the user’s last name, followed by a random number in the range 10 to 99.
- Assume that the last name is at least five letters long.
- Similar algorithms are sometimes used to generate usernames for new computer accounts.
- For example: If the user enters Wayne and Gretzky, your code should print to the console WGretz58 (where 58 could be any random number between 10 and 99)

Exercise

- Write some JavaScript that creates and prints a random phone number of the form XXX-XXX-XXXX
- Include the dashes in the output
- Do not let the first three digits contain an 8 or 9 (but don't be more restrictive than that)
- Make sure that the second set of three digits is not greater than 742
- Hint: Think through the easiest way to construct the phone number. Each digit does not have to be determined separately.