

# CONDITIONALS

- If statement
  - The if(...) statement evaluates a condition in parentheses and, if the result is true, executes a block of code.
- Else statement
  - The if statement may contain an optional “else” block. It executes when the condition is falsy.
- Else if Ladder
  - Sometimes, we’d like to test several variants of a condition. The else if clause lets us do that.

# DIALOGS

- There are basically three types of dialogs that we will use in JavaScript, namely :-
  - Alert
    - It shows a message and waits for the user to press "OK".
    - `alert("You are directing to Website");`
  - Confirm
    - The function confirm shows a modal window with a question and two buttons: OK and Cancel.
    - The result is true if OK is pressed and false otherwise.
    - `confirm("Are you 18 years old");`
  - Prompt
    - `prompt("Enter your age", [default]);`
    - It shows a modal window with a text message, an input field for the visitor, and the buttons OK/Cancel.



# LOOPS

- Loops are a way to repeat the same code multiple times.
- There are 3 basic loops :-

- While

```
while (condition) {  
    // code }
```

- Do...while

```
do {  
    // loop body  
} while (condition);
```

- For

```
for (begin; condition; step) {  
    // ... loop body ... }
```

# LOOPS – CONTINUE & BREAK

- Normally, a loop exits when its condition becomes falsy.
- But we can force the exit at any time using the special break directive.
- The continue directive is a “lighter version” of break. It doesn’t stop the whole loop. Instead, it stops the current iteration and forces the loop to start a new one (if the condition allows).



# FUNCTION

- Functions are the main “building blocks” of the program. They allow the code to be called many times without repetition.
- To create a function we can use a function declaration.

```
function name(parameter1, parameter2, ... parameterN) {  
    ...body...  
}
```

# FUNCTION VARIABLES

- Local variables
  - A variable declared inside a function is only visible inside that function.
- Outer Variables or Global Variables
  - Variables declared outside of any function.
  - Global variables are visible from any function.
  - If a same-named variable is declared inside the function then it *shadows* the outer one.



# FUNCTION PARAMETERS

- We can pass arbitrary data to functions using parameters.
- When a value is passed as a function parameter, it's also called an *argument*.

```
function name(parameter1, parameter2, ... parameterN) {  
  ...body...  
}  
name(2,"String",true)
```

- Default Parameters
  - If a function is called, but an argument is not provided, then the corresponding value becomes undefined.
  - To avoid this we use default parameter for function.
  - `function getName(name="No name Provided") {...}`

# FUNCTION RETURN VALUE

- A function can return a value back into the calling code as the result.
- There may be many occurrences of return in a single function.
- Syntax:-

```
function name(...){  
    return "Hello";  
}
```