

Complete JavaScript

[3 Jan to 10 Jan]

S1:

[3 Jan]

→ Values and variables in JS

key name value
Var myName = 'Tamanna';

console.log(myName);

S2:

Data types in JS

Undefined

Boolean

Number

String

BigInt

Symbol

1! Interview Ques

? Difference between Null vs

Undefined ?

Var iAmUseless = null; → null

Var iAmStandBy; → undefined.

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

2: Interview Ques

What is NaN

2 strings minus + will convert output

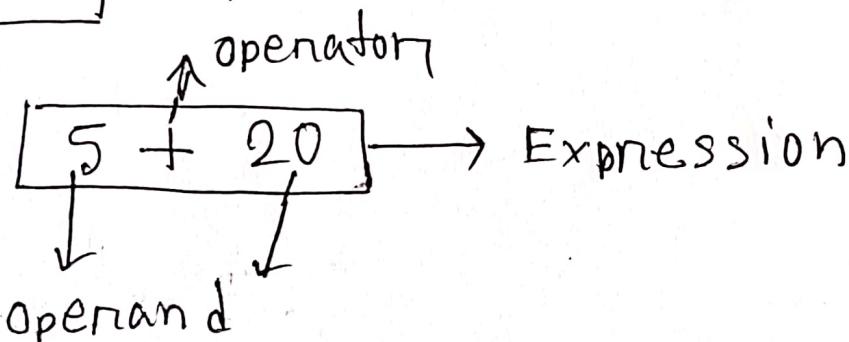
NaN and i

NaN is not a number.

NaN is a global object, Global Scope.

S 3:

Expressions and operations



Types of Expressions & operations

→ Assignment operations $\rightarrow \text{var } n = 5;$

→ Arithmetic $\hookrightarrow (+, -, /, *, \%)$

→ Comparison $\hookrightarrow (=, !=, >, <, !=, !<)$

→ Logical $\hookrightarrow \{\text{AND}(\&&), \text{OR}(||), \text{Not}(!)\}$

→ String $\hookrightarrow \text{Concatenation} (+)$

→ Conditional (ternary) \hookrightarrow

3: I.Q.

Difference between == vs ===?

== → checks only value

==== → checks also the type of the Data.

End → 2:04:18

4 Jan 2022

S4:

Control Statement & Loops

→ If...else... loop as practice

5 falsy values → 0, "", undefined, null, NaN

→ Conditional operator

The conditional (ternary) operator is the only JavaScript operator that takes three operands.

Van age; (age>18)

If true

↑

variable name = (condition) ? value1 : value2

If false

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

→ Switch Statement

Evaluates an expression, matching the expression's value to a case value, and executes statements associated with that case.

→ while Loop

The while loop statement creates a loop that executes a specified statement as long as the test condition evaluates to true.

→ Do while

1st time output (then) check the condition.

→ For Loop

```
for (initialize ; condition ; iteration) {  
    // Code to be executed  
}
```

3:16:01

6 Jan 2022

S5: Functions in JavaScript

①

→ Function definition

```
function functionName(parameters) {  
    // Function body  
}
```

②

→ Calling a function

③

→ Function parameters

④

→ Function arguments

⑤

→ Function expressions

⑥

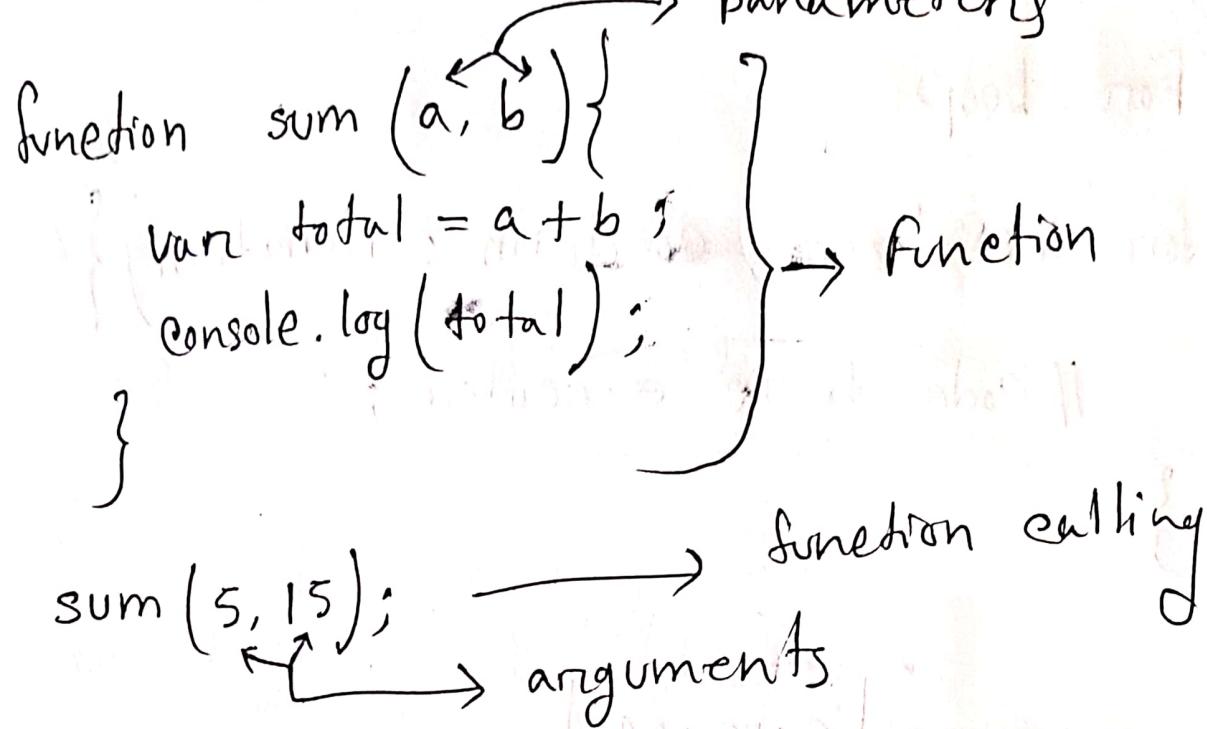
→ Return key word

⑦

→ Anonymous functions

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

1 + 2 + 3 + 4



* Function parameters Vs arguments

- = Function parameters are the names listed in function definition
- = Function arguments are the real values passed to the function while calling a function

* Why functions?

// To reuse code; Define once and use it many times.

= Use the same code many times with different arguments to produce different results.

DRY → Do not Repeat Yourself.

6

```
function sum(a, b) {  
    let total = a + b;  
    return total;  
}
```

Return keyword

```
var funExp = sum(5, 15);  
function expression
```

```
console.log('Total' + funExp);
```

7

```
var funExp = function(a, b){  
    let total = a + b;  
};  
anonymous  
function  
expression
```

```
console.log(funExp(15, 15));
```

```
var funExp = function(a, b) {  
    return total = a + b;  
}
```

```
var sum = funExp(15, 15);
```

```
console.log('Total sum is' + sum);
```

```
var sum1 = funExp(20, 20);
```

```
var sum2 = funExp(20, 40);
```

```
console.log(sum2 > sum1);
```

S6:

ECMA Script

1996 - JavaScript created

1997 - ECMA Script became standard

2015 - ES6

ECMA Script 2015 / ES6

- ① → LET and CONST
- ② → Template strings (Template literals)
- ③ → Default arguments
- ④ → Destructuring
- ⑤ → Object properties
- ⑥ → Arrow function
- ⑦ → Rest operators
- ⑧ → Spread operators

1 Let & const

* Difference Van Vs Let Vs Const

// van → function scope

// let and const → Block scope

4:04:00

9 Jan 2022

2 Template literals

Instead of concatenation

console.log(tableof + "*" + num + " = " + tableof * num);

console.log(` \${tableof} * \${num} = \${tableof * num}`);

3 Default parameters

```
function mult(a, b=5){  
    return a*b;  
}
```

```
console.log(mult(3));
```

5

Fat Arrow Function

// Normal way of writing function

```
console.log(sum());
```

```
function sum(){  
    let a=5, b=6;  
    let sum = a+b;
```

```
    return `The sum is ${sum}`;  
}
```

// Convert it into fat arrow function

```
const sum = () => {
```

```
    let a=5, b=6;
```

```
    let sum = a+b;
```

```
    return `the sum is ${sum}`;  
}
```

Arotril

Clonazepam USP 0.5, 1 & 2 mg Tablet

```
console.log(sum());
```

In fat arrow function you have to first define it then you can call it.

we can simplify the code into?

```
const sum = () => `the sum is ${a=5} + ${b=6}`;
```

```
console.log(sum());
```

S7:

Arrays in JavaScript

Lower Index/boundary
↑
`var name = ["Tamanna", 26, "Islam", True];`
↓

1st index num = 0

Open Index
boundary

Last

`: array.length - 1`

- Traversal of an Array
- Searching and filter in an Array
- How to sort and compare an Array
- How to Insert, Add, Replace and Delete Elements in Array (CRUD)
- Map(), Reduce(), Filter()

→ → Traversal in array ← ←

var myFriends = ['vinod', 'ramesh', 'tanjun',
'vishal'];

console.log(myFriends[1]);

console.log(myFriends[myFriends.length - 1]);

For Loop for navigate

For in → we get index number

for of → n^o elements

for each → (n^o) element, index, array

Traditional :-

```
var myfriends = ['Vinod', 'Ramesh', 'Anjun', 'Tnine'];
```

```
for (var i=0; i<myfriends.length; i++) {  
    console.log(myfriends(i));  
}
```

for In :- (index, print)

```
for (let element in myfriends) {  
    console.log(element);  
}
```

for of :- (elements print)

```
for (let element of myfriends) {  
    console.log(element);  
}
```

For each :-

```
myfriends.forEach(function(element, index, array) {  
    console.log(element, index, array);  
});
```

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

For each in Fat array function :-

```
myFriends.forEach = ((element, index, array) => {  
    console.log(element, index, array);  
});
```

4:55:15

Searching & filter in Array

Searching methods :-

⇒ indexOf()

Elements of array

arrayName.indexOf("String", Number):

↳ Index num
forward search

⇒ lastIndexOf()

Same as indexOf but it does

backward search.

$\Rightarrow \text{includes}()$ - case sensitive

- forward search

$\{200, 300, 400, 500, 600\} \cdot \text{includes}(400)$

$\rightarrow \text{Parameters (String, Number)}$

$\text{const values} = \{100, 200\} \cdot \text{includes}(200)$

--- filtration in Array -

① find() :-

$\Rightarrow \text{arrayName}.find(\text{value}, (\text{index}, \text{array}))$

(Object creation from the value of array) :-
 $\text{const pniees} = [200, 300, 400, 500, 600, 700]$

// pniee < 400 by filter to return object

$\text{const findElem} = \{\text{pnies}\}.find((\text{currVal}) \Rightarrow \{$

return currVal < 400;

\}; // function exp of fat array func

`console.log(findElem);`

Result : 200

Problem is it will return only **Arotrol**
1 value.

Clonazepam USP 0.5, 1 & 2 mg Tablet

Simplify the code :-

const prices = [200, 300, 400, 500, 600, 700];

console.log(prices.find((currVal) => currVal < 600));

→ if condition doesn't satisfy then
return undefined.

⇒ **findIndex()**

→ find the only element like **find()**
function but this time return the
index num of that value.

→ if condition doesn't satisfy then
return -1;

(findIndex gives index of element which
satisfies the condition)

Example :-
const products = [{id: 1, name: "Laptop", price: 1000}, {id: 2, name: "Mouse", price: 200}, {id: 3, name: "Keyboard", price: 300}, {id: 4, name: "Monitor", price: 400}, {id: 5, name: "Headphones", price: 500}]

8908, 06/17
DO: 08:00

⇒ filter()

// Returns a new array containing all elements of the calling array for which the provided filtering function returns true.

filter uses left recursion because it needs to

-----Sort and compare array-----

-----Sort-----

The sort() method sorts the elements of an array in place and returns the sorted array.

The default sorting order is ascending, build upon converting the elements into strings.

However if numbers are sorted as strings "25" is bigger than "100" because $2 > 1$.

Because of the sort() method produces incorrect result for sorting numbers.

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

17 Jan, 2022

5:30:00

push Vs, unshift

→ adds elements at first of the array & returning length.

→ adds elements on last. of the array and returning the new length of the array.

Pop → pop() method removes the last element from any array and returns that element.

shift → removes the first element and returns that removed element.

print statements are separated by new line.
Ex & output "Hello" first print statement
is printed on the first line.

but if first print statement is not present then it will print nothing.

⇒ Complete CRUD operation using
only 1 method.

④ Splice method

takes 3 parameters

1. Starting index (num) no
2. counts Delete counts
3. String New

splice method returning the deleted elements as a linear array.

pairings of parents & children
has no children and has 3
with siblings same as to parents
and in breadth first search

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

\Rightarrow Map(), Reduce(), Filter()

→ Map()

- Returns a new array containing the result of calling a function on every element in this array.

Why use map() instead of forEach()?

* The main difference between map and forEach is that the map method returns a new array by applying the callback function on each element of an array while the forEach method doesn't return anything.

* You can use `forEach` method to mutate the source array but it isn't really the way it's meant to be used. Instead, it's great for when you need to do some action with each element of the array.

On the other hand `map` method is used for creating a new array and thus, it's chainable. You can call a number of `map` operations one after the other.

The `forEach` method doesn't return anything so it's not chainable.

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

Which one should I use, map or forEach?

If you are planning to alter the array elements by applying a function, you should use the map method, since it doesn't modify the original array and returns a new array. In this way the original array is kept intact. On the other hand if you want to loop through all the elements of an array and doesn't return a new array use the forEach method.

⇒ Reduce()

- Reduce method flatten an array means to convert the 3d or 2d array into a single dimensional array.
- The reduce() method executes a reducer (that you provide) on each element of the array, resulting in single output value.

The reduce function takes 4 arguments :

- Accumulator
- Current value
- Current Index
- Source Array

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

- Strings in Java Script -

- Escape Characters
- Finding a string in a String
- Searching for a string
- Extracting String parts
- Replacing string content
- Extracting n strings
- Other useful methods

A. String methods

1. length method

2. charAt() method

3. slice() method

4. replace() method

5. join() method

6. split() method

⇒ Extracting String parts

— slice() method extracts a part of a string and returns the extracted part in a new string.

Takes 2 parameters

— The first parameter (index num) specifies the start position (does not include the given argument)

— The second parameter (index num)

— substring() method is similar to slice() method. The difference is that substring() method can not take negative indexes.

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

⇒ Replacing String Content

replace() replaces a specified value with another value in a string.

- .replace(searchFor, replaceWith)

(function) replace() method doesn't change the original string it is called on. It returns a new string.

- By default the replace() method replaces only the first match.

(parameter) CaseSensitive (optional)

Boolean (optional) Default is false

, specifying whether or not

→ Extracting String Characters

There are (3) methods for this -

- ① charAt (position)
- ② charCodeAt (position)
- ③ PropertyAccess []

#. charAt () method returns the character at a specified index (position) in a string

charCodeAt () method returns the unicode of the character at a specified index in a string. The method returns a UTF-16 code (an integer between 0 and 65535).

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

Property Access

- ECMAScript 5 (2009) allows property access `[]` on strings.

S8: Date and Time in JavaScript

- Date Methods (get & set)
- Time

⇒ Creating Date object

There are 4 ways

- `new Date()`
- `new Date(year, month, day, hours, minutes, seconds, milliseconds)`
- `new Date(milliseconds)`
- `new Date(date string)`

⇒ Date Methods

const curDate = new Date();

// how to get individual date

console.log(curDate.toLocaleString());

console.log(curDate.getFullYear());

console.log(curDate.getMonth()); // 0-11 Jan to Dec

console.log(curDate.getDate());

console.log(curDate.getDay());

* to work on specific month (c)

(addition) methods with combination (c)

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

S9:

JavaScript Math object

Properties ↗ class ↗ properties

→ Math.PI()

→ Math.round()

returns the value of x rounded to its nearest integer.

→ Math.pow() // (n, y)

returns the value of x to the power of y .

→ Math.sqrt()

(n) returns the square root of x .

→ Math.abs()

(n) returns the absolute (positive) value of x

- **math.ceil()**
(n) returns the value of x rounded up to its nearest integer.
- **math.floor()**
opposite of ceil
- **math.min()**
Used to find the lowest value in a list of arguments.
- **math.max()**
Find the highest value
- **math.random()**
returns a random number between 0 (inclusive), and 1 (exclusive)
- **math.trunc()**
returns the integer part of a number.

S10:

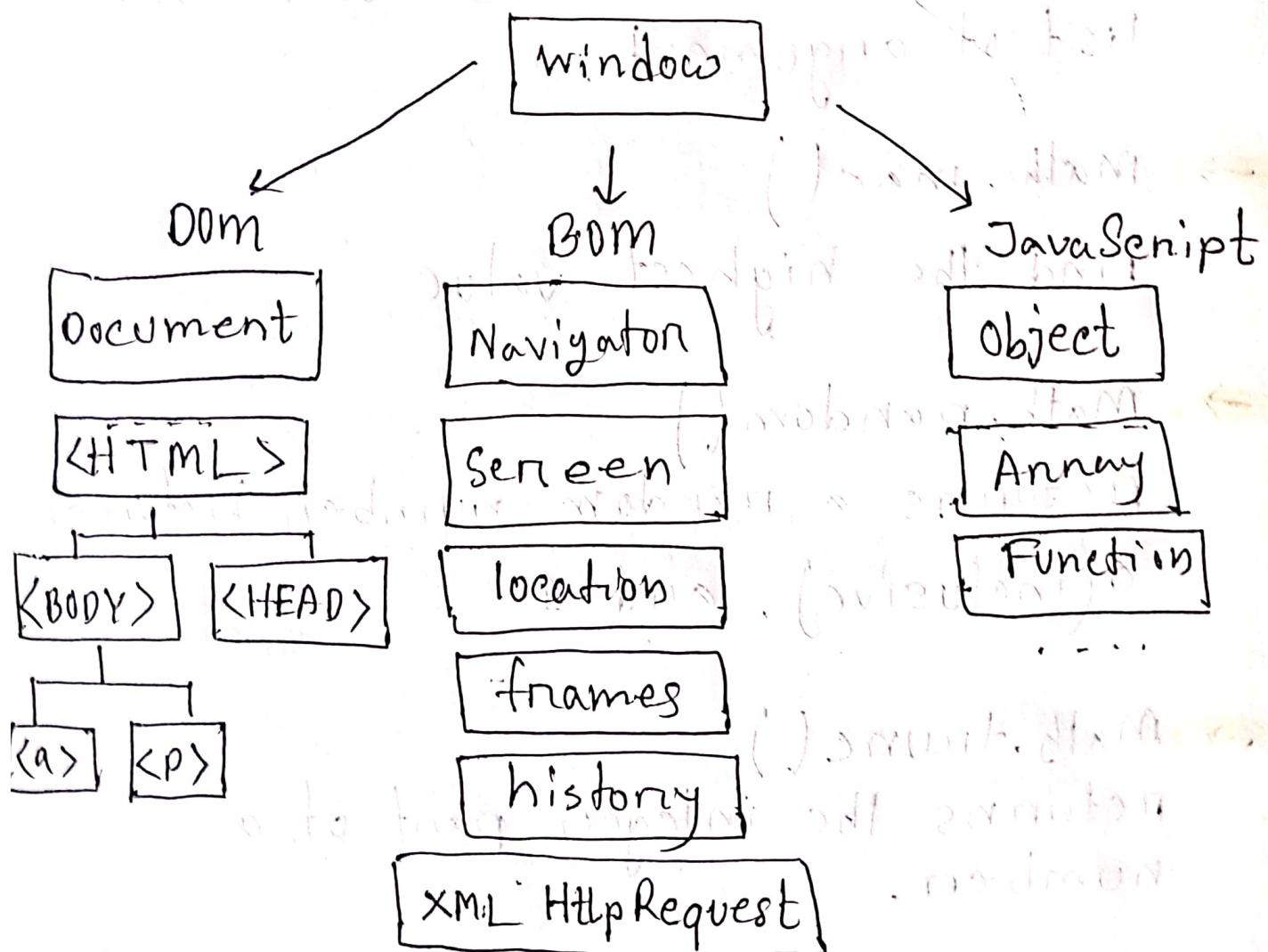
DOM in JavaScript

→ Window Vs Document

→ DOM Vs BOM

→ DOM Navigation

→ Searching and getting Elements Reference



⇒ DOM

- DOM is the Document object model, which deals with the document, the HTML elements themselves, e.g. document and all traversal you would do in it, events, etc.

for Ex: change the background to red.
document.body.style.background = "red";

⇒ BOM

- BOM is the Browser Object Model which deals with browser components aside from the document, like history, location, navigation and screen (as well as some others that vary by browser). OR in simple meaning all

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

the window operations which comes under BOM are performed using BOM.

24 Jan 22

Navigate through the DOM

1. document.documentElement

// returns the element that is the root element of the document.

2. document.head

3. document.body

4. document.body.childNodes (include tab, enter and whiteSpace)

// list of the direct children only.

5. document.children (without text nodes, only regular Elements)

C. document.childNodes.length

practice Time

→ HTC whether an Element has child nodes or not ?

— use hasChildNodes()

→ find the child in DOM tree with

— firstChild Vs firstElementChild

text ↓
actual element ↓

→ lastChild Vs lastElementChild

→ How to find parentNodes

— document.body.parentNode

— document.body.parentElement

→ Find on access the siblings

- document.body.nextSibling
- document.body.nextElementSibling
- document.body.previousSibling
- " ". previousElementSibling

-- How To Search the elements & References

Search and change method

document.getElementById('heading').innerHTML

= "Welcome to thapa";

By taking the reference, heading is the object hence heading.innerHTML = "welcome to thapa";

const headingChange = document.getElementById('heading');

headingChange.innerHTML = "welcome to thapa";

IQ: Difference between getElementById

and querySelector?

getElementById() Syntax:-

element = document.getElementById(id);

// Returns a reference to the Element by its ID. If the element with specified ID is not in the document, it will return null.

querySelector Syntax :-

element = document.querySelector(selectors);

// Returns the first element within the document that matches the specified group of selectors, or null if no matches are found.

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

S:11 Events in JavaScript

// HTML events are things that happen to HTML elements.

// When JavaScript is used in HTML pages, JavaScript can react on these events.

// 4 ways of writing Events in JavaScript

1: Using alert()

2: By calling a function

3: Using Inline events (HTML onclick)

4: Using Event Listeners

onclick same event overrides previous ignore the first call and

Onclick Vs EventListener

Onclick → If we call same events with same reference then it overrides the top event and show the bottom one.

Event Listener → We can call same events multiple times.

⇒ Event Object

- Event object is the parent object of the event object.

Ex: MouseEvent, focusEvent, keyboardEvent etc.

⇒ MouseEvent in JavaScript

⇒ KeyboardEvent in JavaScript

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

⇒ Input Events in JavaScript

- The onchange event occurs when the value of an element has been changed.
- For radio buttons and checkboxes, the onchange event occurs when the checked state has been changed.

IQ: InLine Event Vs EventListener

⇒ addEventListener does not ~~and~~ overwrite existing events handlers.

whereas onclick overrides any existing ~~on~~click = fn event handlers.

The other significant difference, of course, is that onclick will always work, whereas addEventListener does not work in Internet Explorer before version 9.

10:30:57 - - - - - 28 Jan 22

S11:

Timing Based Events in JS

The 2 key methods to use with JS

: one & a function is bound to it

① setTimeout (function, milliseconds)

// Executes a function, after waiting a specified number of milliseconds.

② setInterval (function, milliseconds)

// Same as setTimeout(), but repeats the execution of the function continuously.

both of which must be stopped

or else it will loop forever

Arotril
Clonazepam USP 0.5, 1 & 2 mg Tablet

S12: OOP in JavaScript

→ Object Literal

- Object literal is simply a key:

value pair data structure.

String variables and functions

together in one container,

we can refer this as an objects.

IQ: When I mention "this" in object

what happens?

- Nothing just change the current context from window to that current object where we use it.