

Functions

A JavaScript function is a block of code designed to perform a particular task.

Example:

```
function _name(){  
    alert('hi javascript')  
}
```

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

Return in Func

```
const myFunction = new Function("a", "b", "return a * b");  
  
document.getElementById("demo").innerHTML = myFunction(4, 3);
```

```
document.getElementById("demo").innerHTML = para(4, 3);  
  
function para(x, y = 2) {  
    return x * y;  
}
```

Var Global and Local

Func Date() and practice

recursion function : (The recurse() function is a return function that calls itself inside the function code. The return function must have a stop condition so that it does not call itself again)

```
function recurse() {  
    // statemet....  
    recurse();  
    // statement...  
}  
recurse();
```

```
function recurse() {  
    if(condition) {  
        recurse();  
    }  
    else {  
        // stop  
    }  
    recurse();  
}
```

invoked or anonymous

```
(function () {  
  document.getElementById("demo").innerHTML = "Hello! I called myself";  
})();
```

This in function and addEventListener

Another

```
function myFunction(a, b) {  
  return arguments.length;  
}
```

addEventListener() method => Continue DOM

```
document.getElementById("para1").addEventListener("click", myFunc);  
  
element.addEventListener("click", function(){ alert("Hello World!"); });  
  
window.addEventListener("resize", function(){  
  
  document.getElementById("demo").innerHTML = Math.random();  
  
});  
  
document.getElementById("para1").removeEventListener("mousemove", myFunction);
```

Event bubbling and capturing

Propagation refers to how events travel through the Document Object Model (DOM) tree!

```
select.addEventListener('click', (event) => {  
  .....  
}, false)  
  
Or  
  
select.addEventListener('click', (event) => {  
  .....  
}, true)
```

Stop propagation:

```
<p>Click DIV 1:</p>

<div onclick="func2()">DIV 2

  <div onclick="func1(event)">DIV 1</div>

</div>

<script>

function func1(event) {

  alert("DIV 1");

  event.stopPropagation();

  or:

  event.stopImmediatePropagation()

  or:

  event.cancelBubble = true

}

function func2() {

  alert("DIV 2");

}
```

preventDefault()

```
document.getElementById("myLink").addEventListener("click", function(event){

  event.preventDefault()

});
```

HTML DOM Events

- change
- click
- dblclick
- copy
 - navigator.clipboard.writeText(copyText.value);
- paste
- cut
- focus
- focusin
- focusout
- fullscreenchange
- input => online write
- keydown
- keypress
- keyup
- mousedown
- mouseenter
- mouseleave
- mousemove
- mouseover
- mouseout
- mouseup
- wheel
- play
- stop()
- resize
- scroll
 - window.scrollTo(200, 150);
 - Scroll to down => onclick="scrollWin(0, 150)"
 - Scroll to up => onclick="scrollWin(0, -150)"
 - function scrollWin(x, y) {
 - window.scrollTo(x, y);
 - }
 - alert("pageXOffset: " + window.pageXOffset + ", scrollY: " + window.scrollY);
 -
 - const element = document.getElementById("myDIV");
 - let x = element.scrollLeft;
 - let y = element.scrollTop;
 - const element = document.getElementById("content");
 - let y = element.scrollHeight;
 - let x = element.scrollWidth;
- select
- submit
- onload
 -
 - function loadImage() {
 - alert("Image is loaded");
 - }
 - Onloadeddata => for video or audio
 - Onloadedmetadata => for video or audio
 - Onloadstart
 - Onplay
 - Onseeked => for video line
 - onvolumechange

- event.clientX
- event.clientY
- event.code
- event.data
- event.key
- event.keyCode
- offset
 - <div onclick="myFunction(event)" ></div>
 - <script>
 - function myFunction(event) {
 - let x = event.offsetX;
 - let y = event.offsetY;
 - document.getElementById("demo").innerHTML = "The x-coordinate is: " + x + " : " + y;
- pageX
 - <body onclick="showCoords(event)">
 - function showCoords(event) {
 - let x = event.pageX;
 - let y = event.pageY;
 - let text = "X coords: " + x + " , Y coords: " + y;
- preventDefault
- event.screenX
- event.screenY
- dragend
- dragging
- dragleave
- dragover
- dragstart
- beforeunload
- <body onafterprint="myFunction()">
- select.addEventListener("animationstart", "The animation has started");
- select.addEventListener("animationiteration", "The animation was played again");
- select.addEventListener("animationend", "The animation has completed");
- blur => focus out input
- contextmenu => rightClick
- <input type="search" onsearch="myFunction()">
-
- <video controls ontimeupdate="myFunction(this)">
- <source src="mov_bbb.mp4" type="video/mp4">
- </video>
- function myFunction(s) {
- document.getElementById("demo").innerHTML = s.currentTime;
- } => video time

Navigating Between Nodes

- parentNode
- nodeName
- nodeValue
- childNodes[nodenumbr]
- firstChild
- lastChild
- nextSibling
- previousSibling
- insertBefore
- insertAfter

create element

```
const para = document.createElement("p");

const node = document.createTextNode("This is new.");

para.appendChild(node);

const element = document.getElementById("div1");

element.appendChild(para);
```

remove element

```
document.getElementById("p1").remove();
.....

const parent = document.getElementById("div1");

const child = document.getElementById("p1");

parent.removeChild(child);
.....
```

If And Switch

```
if(x
condition){

    /////
}else if(x condition){

    /////
}else{

    ///
}

(age < 18) ? "Too young":"Old enough";
```