

html5= html태그 + css + java script

13장 내장객체 - 브라우저 객체

window

<script>

모든 변수와 함수 window 포함.

window.open() / close() / setTimeout() / setInterval() / alert.../parsexxx

window.document- html 문서 제공(DOM)

window.navigator-브라우저 정보 제공(OS,종류)

==> navigator.userAgent

window.navigator.geolocation – 브라우저 위치 정보 제공(GPS, WIFI, 기지국, 네트워크망)

==> var a = navigator.geolocation.watchPosition(a , b);

function a(위치정보객체){ 위도 경도 }

function b(오류내용){ 코드 메시지}

navigator.geolocation.clearWatch(a);

window.history – 사이트 이동 정보 모아서 기록

history.go(-1) == history.back();

history.go(1) == history.foward();

window.location – 다른 사이트 이동 주소 제공

location.href="주소변경이동"

14장 DOM- html태그를 자바스크립트 객체 변환 모델

var onetag = document.getElementById("id값");

var manytags = document.getElementsByTagName("태그이름");

var manytags = document.getElementsByClassName("클래스속성값");

var manytags = document.getElementsByName("name속성값");=>input태그

var onetag = document.querySelector("css 선택자 문법");

var manytags = document. querySelectorAll("css 선택자 문법");

<script>

//DOM 구성 완료 대기 후 실행

window.onload = funtion(){

var manytags = document.getElementsByTagName("h1");

}

</script>

<h1>

h1 a

h1,a

```
var onetag = document.getElementById("id값");
```

onetag.속성

a태그.href

input태그.href==>x

alert(a태그.href)-->속성 getter

a태그.href = "주소"-->속성 setter

```
var manytags = document.getElementsByTagName("태그이름");
```

manytags[0].속성

다른 태그 생성 포함

```
var onetag = document.getElementById("id값");
```

div태그 가정.

```
var newtag = document.createElement("img");
```

img.src="";

img.setAttribute("src", "");

onetag.appendChild (newtag);

- 태그 내부 문자열 영역 접근

one.innerHTML = "<H1>XXX</H1>";==> XXX 큰 글씨 표현

one.textContent = "<H1>XXX</H1>";==> 텍스트 표현

548p

html요소 위치 - css

브라우저 크기에 모든 문서 내용 다 보여줄 수 없는 경우 세로스크롤바 생성

viewport영역 - 브라우저 크기 보여지는 html 영역

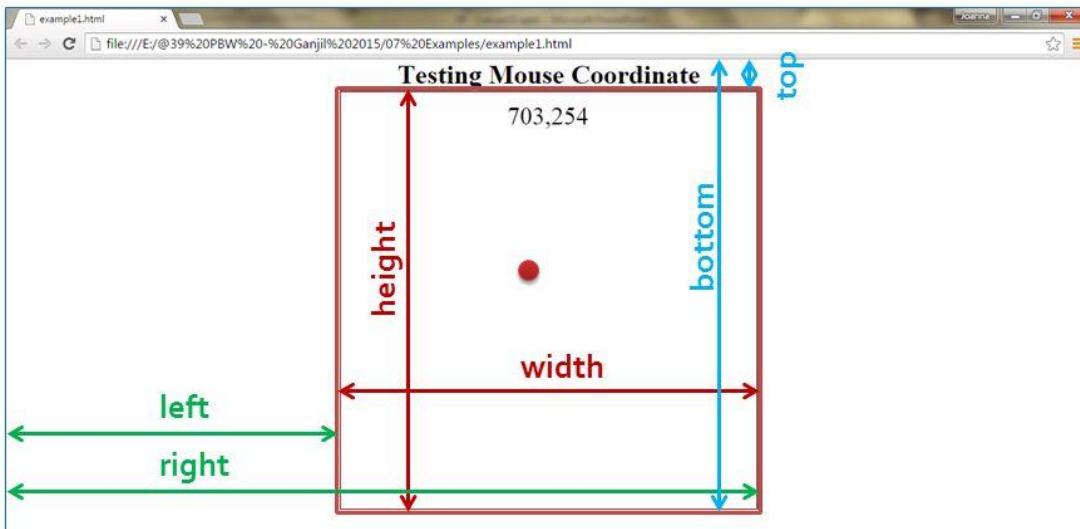
문서영역 - 브라우저 크기 보여지지 않고 가려져 있는 html 영역까지 포함

상단 왼쪽 기준 좌표

x-좌우

y-위아래

Getting Mouse Coordinate



- The **getBoundingClientRect()** method returns the size of an element and its position relative to the viewport

- 특정 태그 1개

```
var onetag = document.getElementById("id값");
```

```
var pos = onetag.getBoundingClientRect();
```

pos.top-> 세로 시작위치

pos.left-> 가로 시작위치

pos.width ->가로크기

pos.height ->세로크기

- 뷰포트 가로크기 / 세로크기

```
document.documentElement.clientWidth
```

```
document.documentElement.clientHeight
```

- 문서 가로크기 / 세로크기(가려진 영역 포함)

```
document.documentElement.scrollWidth
```

```
document.documentElement.scrollHeight
```

- 세로 스크롤 가려진 영역 높이/너비

```
window.pageYOffset/ window.pageXOffset
```



특정 요소로 스크롤

`window.scrollTo(0, 가려져있는 윗부분 + 뷰포트 현재태그 top);`

- html form태그 요소

`<div>`

.....

`<form action="other.jsp" id="f" method=get >`

`<input type=text value="기본값" id="t"`

`<input type=button value="전송" id="b"`

....

`</form>`

`</div>`

- 전송해야 하는 파일명-form태그 action 속성

`alert(document.getElementById("f").action) ;`

==>

`document.forms[0].action`

- 키보드입력 - input text value 속성

```
alert(document.getElementById("t").value) ;  
==> document.forms[0].input[0].value
```

- 버튼 이름이 전송입니까

```
if(document.getElementById("b").value == '전송')
```

```
<h1 >xx</h1>
```

```
var h1= document.getElementsByTagName("h1");
```

```
h1[0].style.color=" blue "
```

```
h1[0].style.border="2px solid red "
```

```
h1[0].style.fontSize="2px solid red "
```

```
_, $
```

```
<h1 style="color:blue;border:2px solid red; ... "
```

15장 이벤트

-이벤트 처리

1. 태그 화면 출력한다

2. 태그 사용자 이벤트 발생한다- 마우스나 키보드

3. 이벤트 후속 동작 처리한다

```
<input ... onclick = "함수호출"
```

```
<input id="a"
```

```
document.getElementById("a").onclick=함수; ==> 무시
```

```
document.getElementById("a").onclick=함수2;
```

```
document.getElementById("a").onclick = null; ==> 이벤트처리x
```

```
<input id="a"
```

```
document.getElementById("a").addEventListener('click', 함수, false);
```

```
document.getElementById("a").addEventListener('click', 함수2, false);
```

```
document.getElementById("a").removeEventListener('click');
```

```
dom요소.onclick = function(){클릭시 실행함수}
```

```
dom요소.onsubmit = function(){submit버튼 클릭시 실행함수}
```

```
dom요소.onkeyup = function(){키보드클릭시 실행함수}
```

```
dom요소.onChange = function(){checkbox, radio 클릭시 실행함수}
```

```
window.onscroll= function(){checkbox, radio 클릭시 실행함수}
```

```
window.onload= function(){checkbox, radio 클릭시 실행함수}
```

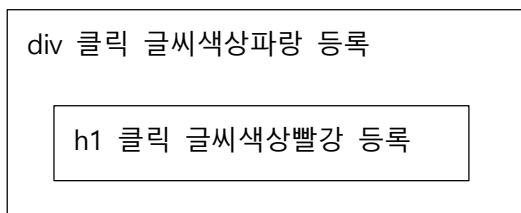
- target - 발생dom요소
 - type - 발생이벤트종류
 - hanlder함수 발생처리핸들러함수
- ==> 모든 종류 이벤트 포함

마우스	target+type+offsetX, offsetY,click, dbclick, mousedown, mouseover
키	keydown , keypressed, keyup,
window	scroll, load
입력	change, submit

- 태그 내부 동작 지정-내장

 간다 간다 클릭하면 b 파일 이동 기본 동작 지정 a태그.onclick = function(){...} + 기본 동작 a태그.onclick = function(ev){ev.preventDefault()};	<form id="f" action="b.jsp"> <input type=submit value="전송"> </form> input submit 값 = 클릭하면 기본 action지정 이 동 기본 동작 docu.... f.onsubmit = function(){...} + action이동 f.onsubmit = function(e){ e.preventDefault()...}
--	---

- 이벤트 전파



1. div 이면서 h1 태그 영역 --> h1(자식) 이벤트 핸들러 함수 실행된다(빨강)
 --> div(부모) 이벤트 핸들러 함수 실행된다(파랑)
 --> 자식태그 이벤트 발생하면 핸들러함수 먼저 실행된다
 이후에 이벤트 부모태그 전달하면 핸들러함수 나중에 실행된다.
 bubbling 전달 막다 - 이벤트객체.stopPropagation()

addEventListener('click', 함수, false(자식->부모 전달));--브라우저 지원

addEventListener('click', 함수, true(부모->자식 전달));
크로스 브라우징 가능 (1개 소스 모든 브라우저 동일 실행 결과 보장)

2. div 이기만 하고 h1 태그는 아닌 영역 --> 클릭 파랑색

- 이벤트 리스너 안의 this

<pre><script> var name="글로벌"; function a(){ var name="로컬"; alert(name); alert(this.name); } window.a(); this instanceof Window --> true</pre>	<pre><script> xx.onclick = b;->this->클릭이벤트target function b(){ this....==> b 함수 핸들러함수 연결 객체 } window.b();--> this-->window</pre>
--	---

react + node(기본지식x)

vue.js + Ajax – 비동기적 처리 + 백엔드(서버) 통신
node.js

18장 클래스
자바스크립트
객체

객체 리터럴값	생성자	ecmascript6 추가 class react
<pre>var obj = { id:xxxx, pw:xxxx, print:function(){ }; 상속표현x var sub = { a:10, obj } --> sub.obj.id</pre>	<pre>function Employee(id, pw){ this.id = id; this.pw = pw; this.print= function(){ --> 새롭게 객체변수 선언 } } var obj1 = new Employee("test1",1234); var obj2 = new Employee("sample", 4321);</pre>	<pre>class Employee{ constructor(id, pw){ this.id = id; this.pw = pw; } print(){ ... } } ///// } //상속 //....</pre>

jquery react