

WEB STUDY 4주차

Javascript with html

목차

➤ Event Handler

➤ Get DOM

➤ 실습

EVENT HANDLER



WHAT IS THE EVENT IN WEBBROWSER

웹 페이지에서는 어떤 종류의 상호작용이 발생할 때 이벤트가 일어난다.

Web-Browser에서 event는 무엇일까?

- ▶ 마우스 클릭이나 키보드 입력 등의 상호작용

브라우저에 의해서 생기는 event 시점

- ▶ 웹 페이지의 로딩이 끝날때

- ▶ 페이지를 스크롤할 때

- ▶ 브라우저 창의 크기를 조절할 때

REGISTER EVENT

```
<button>add one item</button>
```

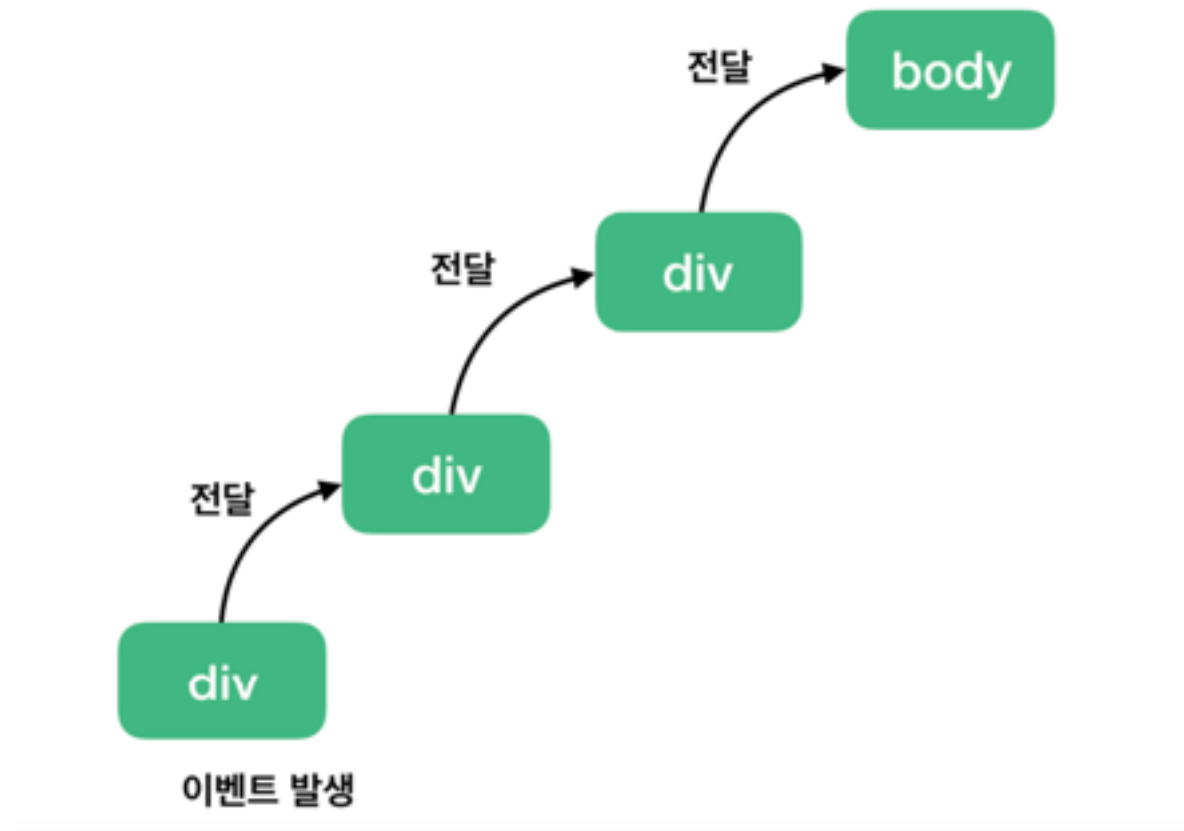
```
var button = document.querySelector('button');  
button.addEventListener('click', addItem);  
  
function addItem(event) {  
    console.log(event);  
}
```

Button 에 *DOM* 객체를 접근하여 이벤트를 등록하는 방식
*Event parameter*는 해당 이벤트의 정보를 담는다.

여기서 *browser*가 이벤트를 감지하는 방식 2가지가 있다.

- *Capture*
- *Bubbling*

EVENT BUBBLING



이벤트 버블링은 특정 화면 요소에서 이벤트가 발생했을 때 해당 이벤트가 더 상위의 화면 요소들로 전달되어 가는 특성을 의미한다.

EVENT BUBBLING

.....

```
<body>
  <div class="one">
    <div class="two">
      <div class="three">
      </div>
    </div>
  </div>
</body>
```

```
var divs = document.querySelectorAll('div');
divs.forEach(function(div) {
  div.addEventListener('click', logEvent);
});

function logEvent(event) {
  console.log(event.currentTarget.className);
}
```

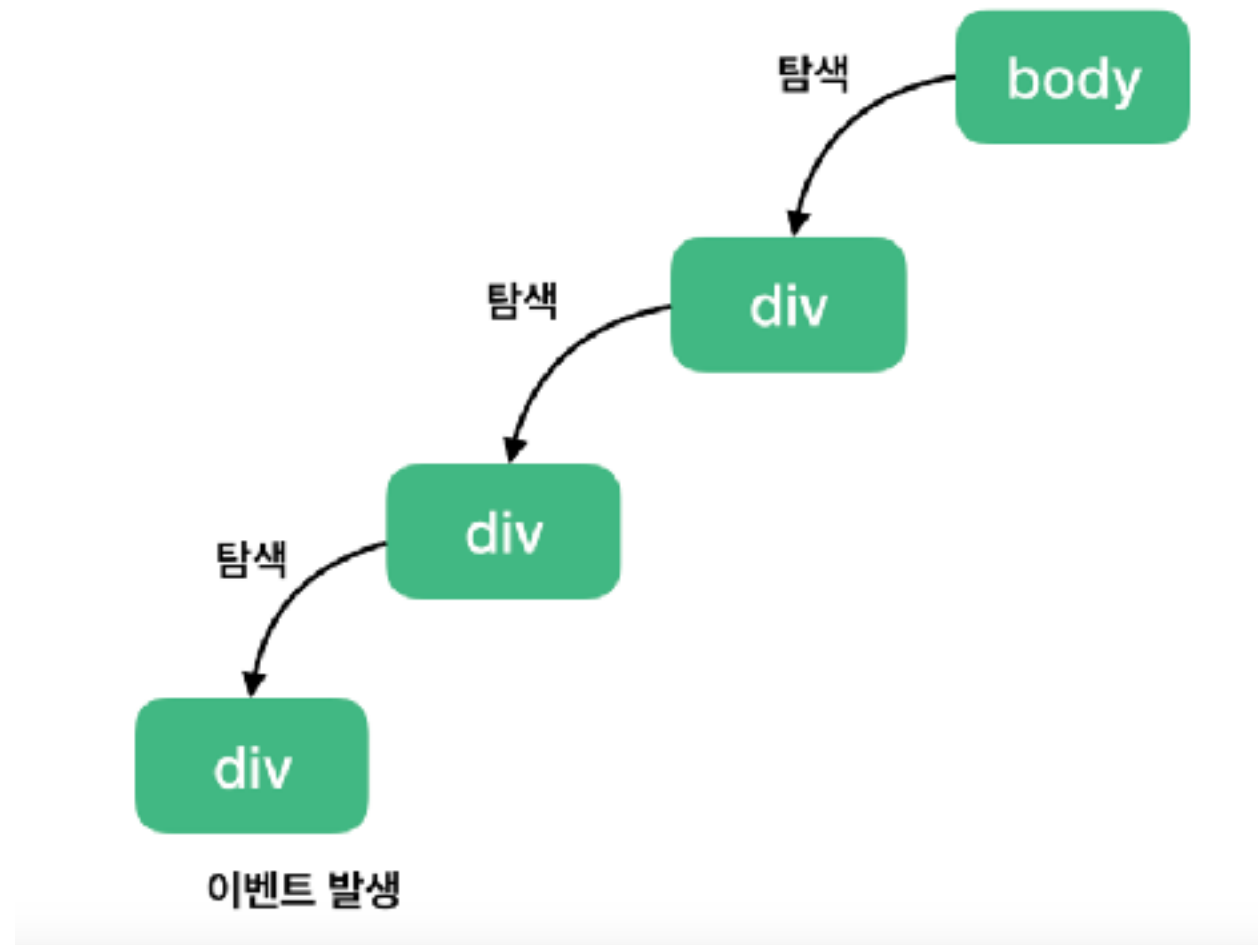
three

two

one

> *div three* 안에서 클릭 이벤트가 발생했을 때 결과

EVENT CAPTURE



이벤트 캡처는 이벤트 버블링과 반대 방향으로 진행되는 이벤트 전파 방식이다.

EVENT CAPTURE

.....

```
<body>
  <div class="one">
    <div class="two">
      <div class="three">
      </div>
    </div>
  </div>
</body>
```

```
var divs = document.querySelectorAll('div');
divs.forEach(function(div) {
  div.addEventListener('click', logEvent, {
    capture: true // default 값은 false입니다.
  });
});

function logEvent(event) {
  console.log(event.currentTarget.className);
}
```

one

two

three

*Capture : true*로 했을 경우 *bubbling*의 반대
방향으로 탐색함

> *div three* 안에서 클릭 이벤트가 발생했을 때 결과

EVENT STOP PROPAGATION

- ▶ 위에 두 방식에서 이벤트가 전파되는 것을 막는 것도 중요하다.
- ▶ 의도치 않은 이벤트 발생을 방지하기 위해서 이벤트 전파를 막을 수 있다.

```
function logEvent(event) {  
    event.stopPropagation();  
}
```

```
// 이벤트 버블링 예제  
divs.forEach(function(div) {  
    div.addEventListener('click', logEvent);  
});  
  
function logEvent(event) {  
    event.stopPropagation();  
    console.log(event.currentTarget.className); // three  
}
```

KEY TYPING

```
<body>
<input type='text' />
<body>
<script>
let input = document.querySelector('input');
input.addEventListener('keypress', func);
function func(event){
  console.log(event);
}
</script>
```

```
▶ KeyboardEvent {isTrusted: true, key: "1", code: "Digit1", location: 0, ctrlKey: false, ...}
▶ KeyboardEvent {isTrusted: true, key: "2", code: "Digit2", location: 0, ctrlKey: false, ...}
▶ KeyboardEvent {isTrusted: true, key: "3", code: "Digit3", location: 0, ctrlKey: false, ...}
```

> 123을 입력했을 때 결과

REGISTER EVENT THROUGH HTML TAG

- ▶ HTML tag에서 이벤트를 등록하는 방식이 존재한다.
- ▶ 자바스크립트로 등록하는 것에 비해 매우 간단하다.

```
<body>
<button onclick="func()">click</button>
<body>
<script>
function func(){
    console.log('Hello World!');
}
```

OnClick 속성으로 이벤트를 등록할 수 있다.

REGISTER EVENT THROUGH HTML TAG

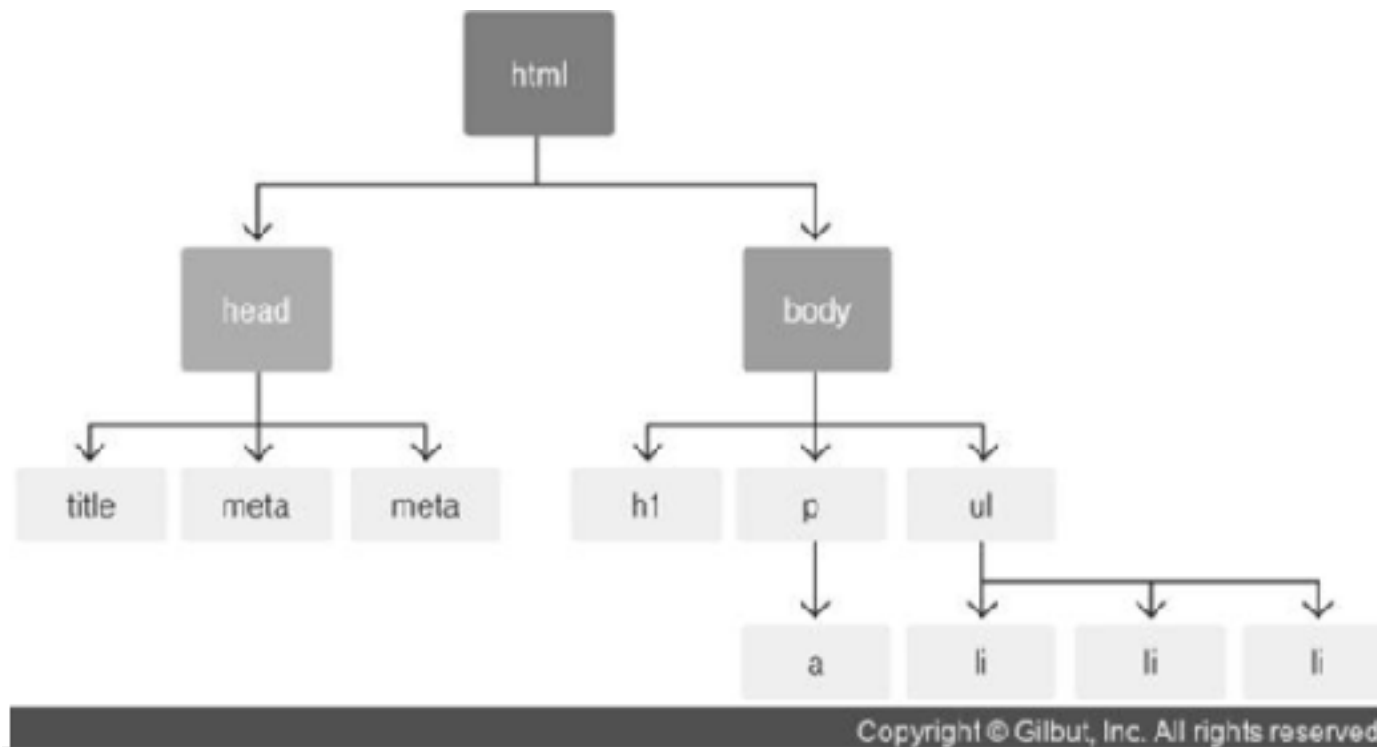
- ▶ Key 입력은 onkeypress 속성을 사용할 수 있다.

```
<input type='text' onkeypress="func()" />
<body>
<script>
function func(){
    console.log('Hello World!');
}
</script>
```


GET DOM



DOCUMENT OBJECT MODEL (DOM)



```
document.getElementsByTagName('body')
```

> javascript code

HTML안에 구성 요소들을 구조적인 object 트리형식으로 구성된 모델이다. 저런 object들은 부모 노드를 식별될 수 있으며, javascript를 통해 document 구조, 스타일, 내용들을 변화시키기 위해 접근될 수 있다.

GET DOCUMENT

Document object를 가져오는 방식

- `document.querySelector()`
- `document.querySelectorAll()`
- `document.getElementById()`
- `document.getElementsByName()`
- `document.getElementsByTagName()`
- `document.getElementsByClassName()`

CHILD & PARENT IN DOM

```
var c = document.getElementById("myDIV").children
```

> 자식 *object reference*

```
var x = document.getElementById("myLI").parentElement
```

> 부모 *object reference*

DOCUMENT PROPERTY

- `document.~.value` : return tag value
- `document.~.checked` : return tag checked value (true | false)
- `document.~.selected` : return tag selected value (true | false)
- `document.~.id` : return tag id
- ...

더 많은 property가 있지만 자주 사용하는 property 중 return 되는 것만 소개하겠습니다.

실습

- ▶ 회원가입할 때 아이디, 비밀번호, 이메일 rule을 만든다.
- ▶ Rule에 맞지 않을 경우 회원가입이 안되도록 한다.
- ▶ 아이디는 영문+숫자 조합 특수문자 불 포함, 7글자 이상 20글자 이하
- ▶ 비밀번호 영문+숫자+특수문자 조합, 10글자 이상 20글자 이하
- ▶ 이메일 part1@part2.domain1.domain2 => .domain2부분은 없어도 된다. Part1,part2,domain1,domain2 에는 특수문자가 포함되면 안되며, 영문+숫자 조합, 각 2글자 이상
- ▶ 이벤트들은 javascript를 이용해서 등록!

HINT

- ▶ AddEventListener('keyup',function) => keyup을 사용 (keypress X)
- ▶ Regular expression을 쓰면 편함
- ▶ document.~.innerText = 'string' => 해당 tag에 text가 작성됨

jjw8967	pass
*****	pass
jjw8967@ajou.ac.kr	pass
submit	

jjw8967!	fail
*****	pass
jjw8967@ajou.ac.kr.	fail
submit	

jjwjjjjj	fail
*****	pass
jjw8967ajou.ac.kr	fail
submit	

jjw	fail
*****	pass
jjw8967@ajou.com	pass
submit	

jjw8967	pass
*****	pass
jjw8967@ajou	fail
submit	

EXAMPLE

jjw8967	pass
.....	pass
jjw8967@ajou.ac.kr	pass
submit	

jjw8967!	fail
.....	pass
jjw8967@ajou.ac.kr.	fail
submit	

jjwjjjjjjj	fail
.....	pass
jjw8967ajou.ac.kr	fail
submit	

jjw	fail
.....	pass
jjw8967@ajou.com	pass
submit	

jjw8967	pass
.....	pass
jjw8967@ajou	fail
submit	