

# **JQUERY**

# 과제1 해설

```
<body>
  <h1>여기가 바뀔거예요</h1>
  <h1 id="result">_____</h1>

  <input type="text" id="input"/>

  <script>
    var result = document.querySelector('#result')
    var input = document.querySelector('#input')

    input.addEventListener('keydown', keydown)

    function keydown(event) {
      if(event.which === 13 && input.value) {
        result.textContent = input.value
        input.value = ''
      }
    }
  </script>
</body>
```

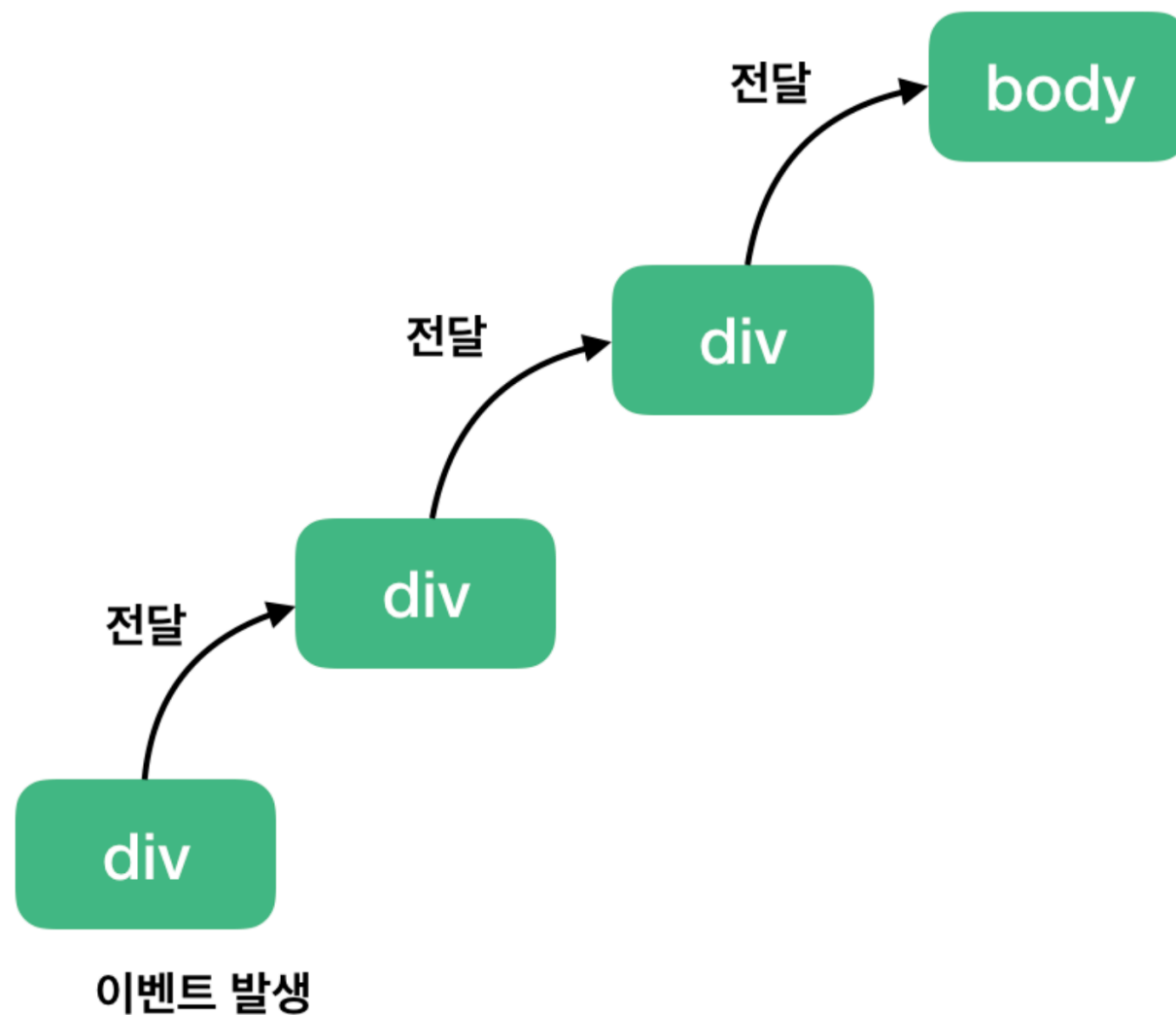
## 과제2 해설

```
<script>

    outside.addEventListener("click", function() {
        console.log("가장 바깥 쪽!");
    });
    center.addEventListener("click", function() {
        console.log("가운데!");
        event.stopPropagation();
    });
    inside.addEventListener("click", function() {
        console.log("가장 안 쪽!");
        event.stopPropagation();
    });
    link.addEventListener("click", function() {
        console.log("구글로 가기!");
        event.stopPropagation();
        event.preventDefault();
    });
</script>
```

# Event Bubbling

- 특정 화면 요소에서 이벤트가 발생했을 때, 해당 이벤트가 더 상위의 화면 요소들로 전달되어 가는 특성



## 과제3 해설

```
<script>
    let divs = document.querySelectorAll("div");

    divs.forEach(function(div) {
        div.addEventListener("click", function() {
            // 누를 때마다 전체 div backgroundColor 초기화
            divs.forEach(function(div) {
                div.style.backgroundColor = "white";
            });

            // 누른 div backgroundColor 변경
            this.style.backgroundColor = "skyblue";

            // parentElement의 tagName이 body가 아니면,
            // backgroundColor 변경
            if (this.parentElement.tagName !== "BODY") {
                this.parentElement.style.backgroundColor = "hotpink";
            }

            // EventBubbling 막기
            event.stopPropagation();
        });
    });
</script>
```

## 과제3 해설

```
<script>
  let divs = document.querySelectorAll("div");

  divs.forEach(function(div, key) {
    div.addEventListener("click", function() {
      // 누를 때마다 전체 div backgroundColor 초기화
      divs.forEach(function(div) {
        div.style.backgroundColor = "white";
      });

      // 누른 div backgroundColor 변경
      this.style.backgroundColor = "skyblue";

      // parentElement의 key가 0이 아니면,
      // backgroundColor 변경
      if (key) this.parentElement.style.backgroundColor = "hotpink";

      // EventBubbling 막기
      event.stopPropagation();
    });
  });
</script>
```

# jQUERY란?

- JavaScript 라이브러리
- HTML Document의 \_\_\_\_\_를 쉽게 만들어준다
  - Selecting / Manipulating / Creating / Animating Elements
  - Add Event Listeners
  - Add Effects
  - Make HTTP Requests (AJAX)

# jQUERY를 왜 쓸까?

## - 장점

- 불완전했던 DOM API
- 사용하기 쉽다.
- 코드가 짧아지고 명확해진다.
- AJAX 사용이 엄청나게 간단하다.
- 거대한 커뮤니티



# 안쓸 이유가 없는데?

## - 장점

- 불완전했던 DOM API
- 사용하기 쉽다.
- 코드가 짧아지고 명확해진다.
- AJAX 사용이 엄청나게 간단하다.
- 거대한 커뮤니티

## - 단점

- 더이상 DOM API가 불완전하지 않다.
- 순수 자바스크립트로 다 할수 있다.
- 몇몇 기능만 사용하기에는 다소 무섭다.
- 거대한 커뮤니티에서 사람들이 빠져나가고 있다.

# 그래서 써요..? 말아요..?

- jQuery는 간단한 프로토타입용 프로젝트를 만들때는 아주 편리한 라이브러리
- 하지만, 확실한 JavaScript에 대한 이해가 밑바탕이 되어야 함
- 즉, jQuery를 적재적소에 잘 활용하되, 너무 의존해서는 안됨

**개발자분**들은 예전에 가장 HOT했던 라이브러리인 만큼, 어떻게 쓰는지 배워두면서,  
JavaScript로는 어떻게 구현할 지 고민해보시고,

**비개발자분**들은 프로토타이핑용으로 배워서 잘 써먹으시면 됩니다.

# jQuery

<http://code.jquery.com/>

## Code Integration

```
<script  
  src="http://code.jquery.com/jquery-3.4.1.min.js"  
  integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="   
  crossorigin="anonymous"></script>
```

The `integrity` and `crossorigin` attributes are used for [Subresource Integrity \(SRI\) checking](#). This allows browsers to ensure that resources hosted on third-party servers have not been tampered with. Use of SRI is recommended as a best-practice, whenever libraries are loaded from a third-party source. Read more at [srihash.org](http://srihash.org)

# 써봅시다!

## JavaScript

## jQuery

```
<script>
  // h1 잡기, 색깔 변경하기
  document.querySelector('h1').style.color = 'red'

  var h1 = document.querySelector('h1')
  var button = document.querySelector('button')

  // button 클릭시 텍스트 바꾸기
  button.addEventListener("click", function(){
    h1.innerHTML = '운장이는 <i>돼지</i>다'
    button.innerHTML = '그거지!'
  })
</script>
```

# 써봅시다!

## JavaScript

```
<script>
  // h1 잡기, 색깔 변경하기
  document.querySelector('h1').style.color = 'red'

  var h1 = document.querySelector('h1')
  var button = document.querySelector('button')

  // button 클릭시 텍스트 바꾸기
  button.addEventListener("click", function(){
    h1.innerHTML = '운장이는 <i>돼지</i>다'
    button.innerHTML = '그거지!'
  })
</script>
```

## jQuery

```
<script>
  // h1 잡기, 색깔 변경하기
  $('h1').css("color", "red")

  var h1 = $('h1')
  var button = $('button')

  // button 클릭시 텍스트 바꾸기
  button.click(function() {
    h1.html('운장이는 <i>돼지</i>다')
    button.text('그거지!')
  })
</script>
```

# jQuery 자주 쓰는 메소드

- text() / html() : html 태그 내부의 text / html 조작
- attr() : src, class, href 등 html 태그의 attribute 조작
- val() : input, select, textarea 등 html 태그의 값 조작
- addClass(), removeClass(), toggleClass() : class 추가 / 제거
- parent(), children(), append(), on() 등등...

# 바로 실습

오늘의 할 일



- 멋쟁이사자처럼 교안 짜기



- 기억심리학 과제하기



# 들어가야 할 기능

- + 버튼 클릭 시, Input field가 fade toggle
- trash icon 클릭 시, 500ms간 fadeout 후 사라짐
- 리스트 누르면 .complete라는 클래스 토글
- 입력 후 엔터 시, 새로운 리스트 생성
- 상단의 trash icon 버튼 클릭 시, 완료된 리스트 모두 삭제



# 들어가야 할 기능

- + 버튼 클릭 시, Input field가 fade toggle
- trash icon 클릭 시, 500ms간 fadeout 후 사라짐
- 리스트 누르면 .complete라는 클래스 토글
- 입력 후 엔터 시, 새로운 리스트 생성
- 상단의 trash icon 버튼 클릭 시, 완료된 리스트 모두 삭제

**HINT :** .fadeToggle() / .on() / .stopPropagation()  
/ .append() / .parent() / this

# 1학기 마지막 과제

## AJAX

**AJAX가 무엇인지 알아보는게 과제!**

# 1학기 마지막 과제

## AJAX

<https://dog.ceo/api/breeds/image/random>

댕댕이 사진 주세요!

멍! 🐾



(+ 1학기 마지막 추가 과제)

## AJAX + Django

오늘의 할 일

- 멋쟁이사자처럼 교안 짜기

- 기억심리학 과제하기