

인터랙션스튜디오: 웹인터랙션프로그래밍



웹인터랙션프로그래밍

ododog12@naver.com

010 9215 5512

나의 일대기를 담은 인터랙티브 사이트 직무수행1

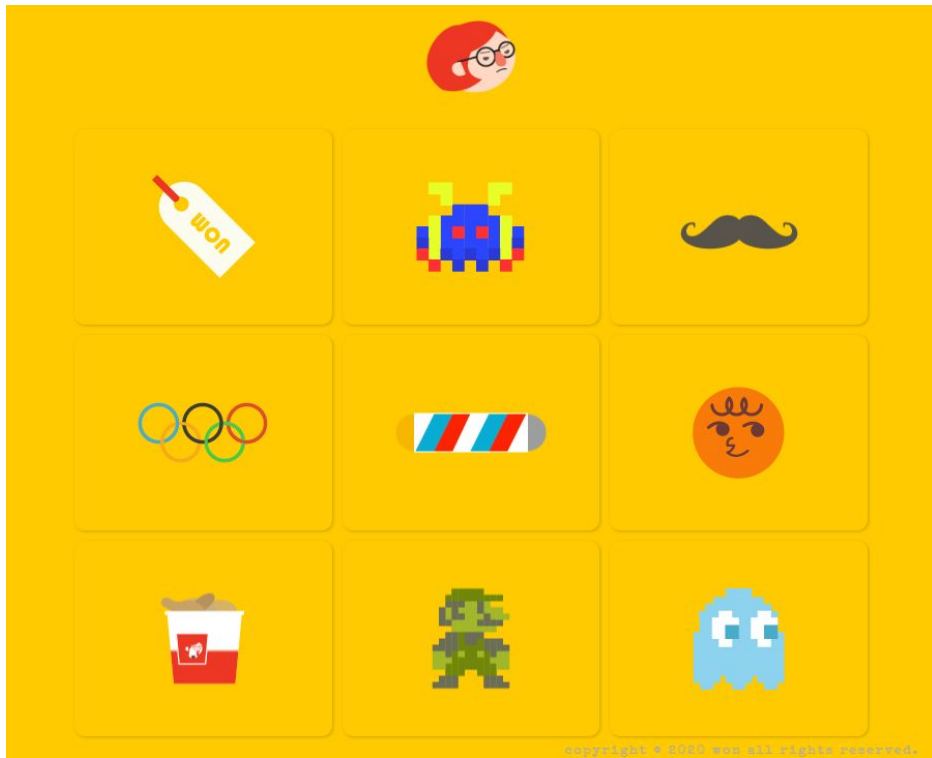
신원	1-3	4-6	7-9	9-11	12-14	15-17	18-20	21-23	24-26
	★	※	◎						
	◆	♠	◐						
	♪	♥	♣						

1. 총 9개의 셀로 구성
2. 메인 화면 구성 자유
3. 반응형 x 화면 1920 고정 > 유동적으로 짜는 것은 환영입니다.
4. 각 셀마다 수업에서 배운 js, 3d, canvas, svg 등 다양한 인터랙션을 적용하여 내용을 구성

이것에 대한 기획서

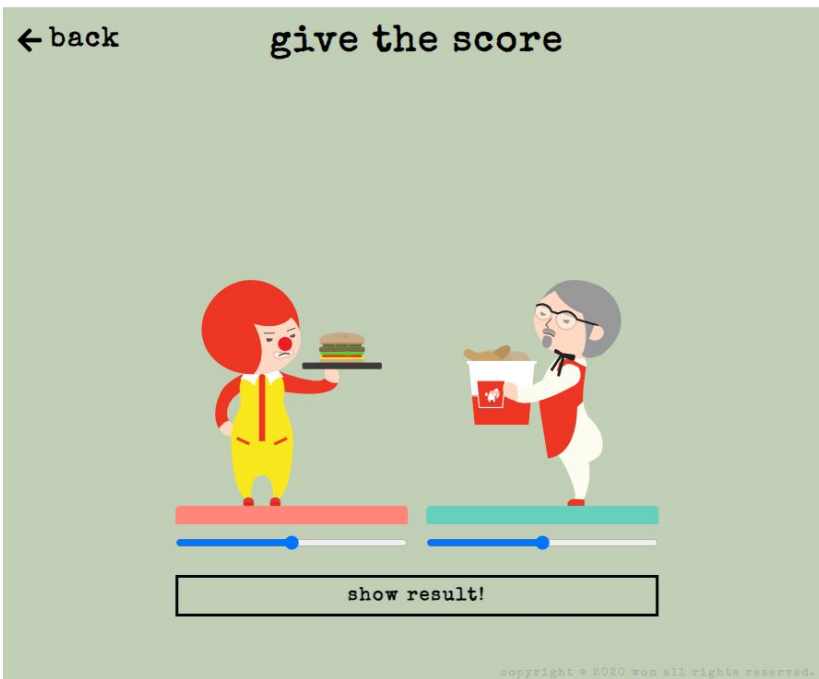
1. 메인구성
2. 각 셀 화면 구성 > 내용 및 계획
3. 디자인 무드

나의 일대기를 담은 인터랙티브 사이트 직무수행2



1. 총 9개의 셀로 구성
2. 메인 화면 구성 자유
3. 반응형 x 화면 1920 고정 > 유동적으로 짜는 것은 환영입니다.
4. 각 셀마다 수업에서 배운 js, 3d, canvas, svg 등 다양한 인터랙션을 적용하여 내용을 구성
5. chatGPT 및 플러그인 사용 지양
(본인이 코드를 구성할 경우 더 높은 평가를 받습니다.)
6. 사용할 경우 기재 및 라이선스 확인
7. 사용자가 직접 인터랙션할 수 있는 방향을 추천

나의 일대기를 담은 인터랙티브 사이트 직무수행2



조건문

if / switch

```
class3
조건문 : if문
```



noahcha

약 1분 전

어느 아내가 프로그래머 남편에게 「쇼핑하러 갈 때, 우유 하나 사와, 아, 계란 있으면 6개 사와」 남편은 잠시 후, 우유를 6개 사왔다. 아내는 물었다. 「왜 우유를 6개나 사왔어!」 남편 「계란이 있길래 6개 사왔지...」

twtr에서 작성된 글

쇼핑하러 갈 때, 우유 하나 사와

계란이 있으면 6개 사와

아내

우유를 하나 사고

만약에 계란이 있으면 계란도 6개 사와



프로그래머 남편

우유를 하나 사

만약에 계란이 있으면 우유를 6개 사



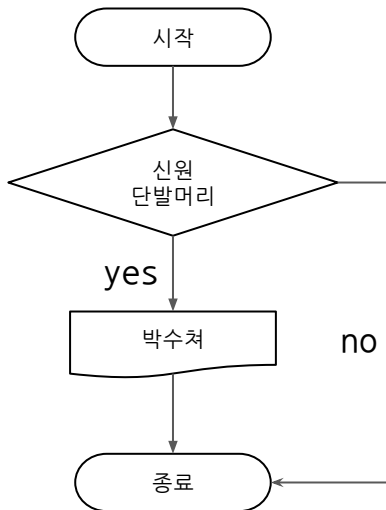
```
class3  
조건문 : if문
```

주어진 조건이 참(true)일때 행동 실행

- if(조건) {행동}
- if(조건) {행동}
else {행동}
- if(조건) {행동}
else if(두번째 조건) {행동}
else {행동}

class3
조건문 : if문

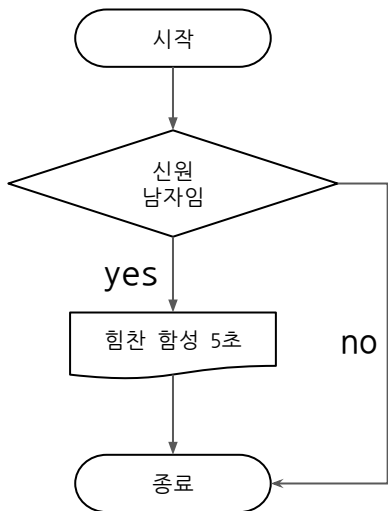
참(true)일때



```
if(신원 단발머리) { function(){  
    박수쳐  
}}
```

class3
조건문 : if문

거짓(false)일때



```
if(신원 남자임) { function(){  
    힘찬 함성 5초  
}}
```

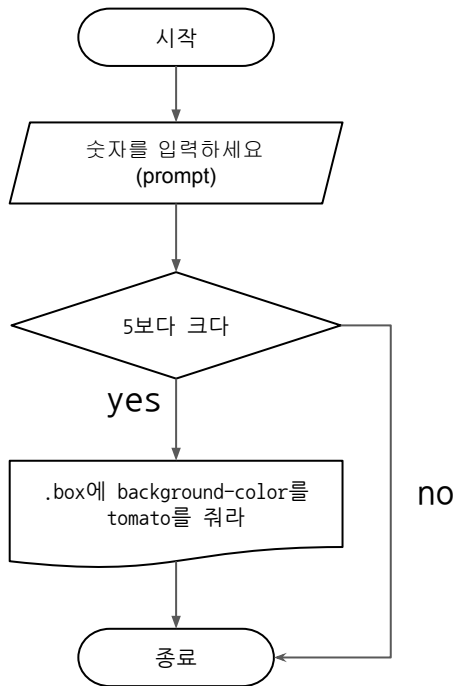
```
class3
```

```
조건문 : if문
```

prompt로 받은 숫자가 ' 5보다 크다 ' 라는 조건이 참(true)일때
.box에 background-color를 tomato로 바꿔주세요

```
class3  
조건문 : if문
```

prompt로 받은 숫자가 ' 5보다 크다 ' 라는 조건이 참(true)일때
.box에 background-color를 tomato로 주세요



class3
조건문 : if문

```
<div class="box"></div>
```

```
<style>
```

```
.box {
```

```
width:100px;
```

```
height:100px;
```

```
background:aqua;
```

```
}
```

```
</style>
```

```
let inputNum = prompt('숫자를 입력하세요');
```

```
inputNum = parseInt(inputNum);
```

```
const box = document.querySelector('.box')
```

```
if(inputNum > 5) {
```

```
    box.style.backgroundColor = 'tomato'
```

```
}
```

실습

```
class3  
조건문 : if문
```

‘ range의 값이 50초과이다 ’ 가 참(true)일때
body의 background를 aqua로 해주고
거짓(false)일때
body의 background를 pink로 해주세요

```
class3  
조건문 : if문
```

element의 value값 가져오기
element.value

class3
조건문 : if문

```
<input type="range">  
<span class="num">50</span>
```

```
const range = document.querySelector('input[type="range"]')  
const num = document.querySelector('.num')  
  
range.addEventListener('change', () => {  
  num.innerHTML = range.value;  
  
  if(range.value > 50) {  
    document.body.style.background = 'aqua'  
  }else {  
    document.body.style.background = 'pink'  
  }  
})
```

```
class3  
조건문 : switch
```

```
switch(케이스) {  
    case ' 케이스1 ' :  
        행동  
        break;  
    case ' 케이스2 ' :  
        행동  
        break;  
    default :  
        행동  
}
```

```
class3  
조건문 : if문
```

라디오를 클릭했을때 value가 cat이면
body의 background를 yellow로 해주고
value가 dog이면
body의 background를 orange로 해주고
value가 ham이면
body의 background를 pink로 해주고

```
class3  
조건문 : switch
```

```
<label>  
  <input type="radio" name="animal" value="cat">고양이  
</label>  
<label>  
  <input type="radio" name="animal" value="dog">개  
</label>  
<label>  
  <input type="radio" name="animal" value="ham">햄  
</label>
```

class3
조건문 : switch

```
const radio = document.querySelectorAll('input[type="radio"]')
```

```
radio.forEach(radio => {
```

```
  radio.addEventListener('click', (e) => {
```

```
    let value = e.target.value
```

```
    switch (value) {
```

```
      case 'cat' :
```

```
        document.body.style.background = 'yellow'
```

```
        break;
```

```
      case 'dog' :
```

```
        document.body.style.background = 'orange'
```

```
        break;
```

```
      case 'ham' :
```

```
        document.body.style.background = 'pink'
```

```
        break;
```

```
    }
```

```
  })
```

```
})
```

```
class3  
조건문 : if vs switch
```

언제 if를 쓰고 언제 switch를 쓸까?

더 짧고 쉽고 이쁜거 써라

배열

array

```
class4  
array 배열
```

배열(Array)은 여러 개의 데이터를 하나의 변수에
저장할 수 있는 데이터 구조입니다.

class4
array 배열

[0]

참지

[1]

루루

[2]

춘봉

```
let cat = ['루루', '춘봉', '참지']  
document.write(cat[0])  
document.write(cat[1])  
document.write(cat[2])
```

class4
array object형

```
<ul>  
  <li>  
    내 이름은 <strong></strong>  
    내 나이는 <span></span>살  
  </li>  
  <li>  
    내 이름은 <strong></strong>  
    내 나이는 <span></span>살  
  </li>  
  <li>  
    내 이름은 <strong></strong>  
    내 나이는 <span></span>살  
  </li>  
</ul>
```

class4
array object형

```
let cat = [  
  {  
    name : '루루',  
    age : 1  
  },  
  {  
    name : '참봉',  
    age : 10  
  },  
  {  
    name : '참지',  
    age : 100  
  }  
]
```

객체 (Object)

- 객체는 이름(키)과 값의 쌍을 가지는 데이터 구조입니다.
- 각 데이터(속성)는 고유한 키를 가지며, 순서는 보장되지 않습니다.
- 객체는 주로 관련된 데이터를 그룹화하는 데 사용됩니다.

class4
array object형

```
const li = document.querySelectorAll('li')

li.forEach((li, index) => {

  li.querySelector('strong').innerHTML = cat[index].name
  li.querySelector('span').innerHTML = cat[index].age
})
```

```
class4  
array
```

push() : 배열의 끝에 요소를 추가

pop() : 배열의 마지막 요소를 제거하고 그 요소를 반환

shift() : 배열의 첫 번째 요소를 제거하고 그 요소를 반환

unshift() : 배열의 앞에 요소를 추가

length : 배열의 길이(요소의 개수)를 반환

```
let arr = []  
  
arr.push('루루')  
console.log(arr)  
arr.push('장군')  
console.log(arr)  
arr.unshift('나나')  
console.log(arr)  
arr.pop();  
console.log(arr)  
arr.shift()  
console.log(arr)
```

4 3 array.html?_ijt=...d=RELOAD_ON_SAVE:12

▶ ['루루']

4 3 array.html?_ijt=...d=RELOAD_ON_SAVE:14

▶ (2) ['루루', '장군']

4 3 array.html?_ijt=...d=RELOAD_ON_SAVE:16

▶ (3) ['나나', '루루', '장군']

4 3 array.html?_ijt=...d=RELOAD_ON_SAVE:18

▶ (2) ['나나', '루루']

4 3 array.html?_ijt=...d=RELOAD_ON_SAVE:20

▶ ['루루']

class4
array

```
let pop = arr.pop();  
console.log(pop)  
let shift = arr.shift()  
console.log(shift)
```

[4_3_array.html?_ijt=...d=RELOAD_ON_SAVE:18](#)

장군

[4_3_array.html?_ijt=...d=RELOAD_ON_SAVE:20](#)

나나

class4
array

```
let arr = ['루루', '나나', '삼색이']
```

```
console.log(arr.length)
```

4_4_array.html?_ijt=.
3


```
class4  
array
```

`splice()` : `array.splice(num, num)`

- 문자열을 특정 구분자를 기준으로 나누어 배열로 반환

`slice()` : `array.slice(start, end)`

- 배열이나 문자열에서 원하는 부분을 추출하여 새 배열이나 새 문자열을 반환

class4
array

```
let arr = ['루루', '나나', '삼색이']
```

```
arr.splice(1, 1)
```

```
console.log(arr)
```

```
let arr2 = ['춘봉', '참지', '덕배', '호동']
```

```
let slice = arr2.slice(1,3)
```

```
console.log(slice)
```

[4_5_array.html?_ijt=...d=RELOAD_ON_SAVE:12](#)

▶ (2) ['루루', '삼색이']

[4_5_array.html?_ijt=...d=RELOAD_ON_SAVE:17](#)

▶ (2) ['참지', '덕배']

반복문

forEach / map

map과 forEach의 차이

차이점 요약

1. **forEach():**
 - 배열의 각 요소에 대해 **반복 작업**을 수행합니다.
 - **반환값이 없습니다**. 원본 배열을 변경하거나, 작업을 수행할 때 사용됩니다.
 - 주로 **DOM** 조작이나, 부수 효과(side effects)를 일으키는 작업에 사용됩니다.
2. **map():**
 - 배열의 각 요소를 변환하고 **새로운 배열을 반환**합니다.
 - 반드시 새로운 배열을 생성하고 그 배열을 사용해야 할 때 사용됩니다.

map과 forEach의 차이

- 이미지 요소를 **li**에 차례대로 출력하는 경우: 이 작업은 단순히 각 배열 요소에 대해 DOM을 조작하는 것이므로 **반환값이 필요하지 않습니다**. 따라서, **`forEach()`**를 사용하는 것이 더 적합합니다.
- **`map()`**은 변환된 배열이 필요할 때 적합합니다. 예를 들어, **새로운 DOM 요소 배열을 생성**하고 나중에 이 배열을 다른 곳에 사용해야 할 때 사용할 수 있습니다 >> **새로운 배열을 생성해서 사용할 필요가 있는지**

class4
반복문 map

```
배열.map((element, index) => {행동});
```

class4
반복문 map

```
<ul></ul>
```

```
const items = ['나나', '도도', '토토']

const liElement = items.map(item => {
  const li = document.createElement('li')
  li.innerHTML = item
  return li
})

const ul = document.querySelector('ul')
liElement.forEach(li => {
  ul.appendChild(li)
})
```

class4

반복문 forEach

```
배열.forEach((element) => {행동});
```


class4
반복문 forEach

```
<style>
  ul {
    display: flex;
    list-style: none;
  }
  li {
    width: calc(100% / 3);
  }
</style>
```

class4
반복문 forEach

```
const img = ['img_1.jpg', 'img_2.jpg', 'img_3.jpg']
const ul = document.querySelector('ul')

img.forEach(url => {
  const li = document.createElement('li')
  const img = document.createElement('img')

  img.src = url
  li.appendChild(img)
  ul.appendChild(li)
})
```

attribute

class3
attribute

element의 value값 가져오기

`element.value`

img의 src값 가져오기

`img.src`

element의 placeholder값 가져오기

`element.placeholder`

checkbox의 checked값 boolean 가져오기

`checkbox.checked (true / false)`

class3
attribute

```
<input type="text" placeholder="이름을 입력하세요">  
<button>로그인</button>  
<h1></h1>
```

class3
attribute

```
const input = document.querySelector('input[type="text"]')
const btn = document.querySelector('button')
const h1 = document.querySelector('h1')

btn.addEventListener('click', () => {
  const name = input.value;
  h1.innerHTML = `<strong>${name}</strong>님 안녕하세요`
})
```

class3
attribute

```
btn.addEventListener('click', () => {  
  const name = input.value;  
  
  if(name !== '') {  
    h1.innerHTML = `<strong>${name}</strong>님 안녕하세요`  
  }else {  
    alert('입력하세요')  
  }  
})
```

class3
attribute

```
btn.addEventListener('click', () => {  
  const name = input.value;  
  
  if(name.length >= 2) {  
    h1.innerHTML = `${name}</strong>님 안녕하세요`  
  }else {  
    alert('입력하세요')  
  }  
})
```


실습

```
class3
attribute
```

```
<div id="wrap">
  <div class="left"></div>
  <div class="right">
    <h1>은혜값은 <strong class="name"></strong></h1>
    <p>옛날 옛날 어느 마을에 <strong class="name"></strong>이 살았어요</p>
    <p><strong class="name"></strong>은 <strong class="age"></strong>살이나 먹은 고양이였어요</p>
    <p>몸무게가 <strong class="weight"></strong>kg나 나가는 똥고양이였어요</p>
  </div>
</div>
<div class="input">
  <input type="text" class="input-name" placeholder="이름을 입력하세요">
  <input type="number" class="input-age" placeholder="나이를 입력하세요">
  <input type="number" class="input-weight" placeholder="몸무게를 입력하세요">
  <button class="btn-submit">출력</button>
</div>
```

class3
attribute

```
* {  
  margin:0;  
  padding:0;  
  box-sizing: border-box;  
}  
#wrap {  
  position: absolute;  
  top:50%;  
  left:50%;  
  display: flex;  
  width:1000px;  
  height:500px;  
  border:5px solid #ccc;  
  transform:translate(-50%, -50%);  
}  
#wrap > div {  
  width:50%;  
}  
#wrap .left {  
  overflow: hidden;  
}
```

```
#wrap .left img {  
  width:100%;  
}  
#wrap .right {  
  padding:50px;  
  font-size:20px;  
}  
#wrap .right h1 {  
  margin-bottom:30px;  
}  
#wrap .right p {  
  line-height:40px;  
}
```

class3
attribute

```
const nameBox = document.querySelectorAll('.name')
const ageBox = document.querySelector('.age')
const weightBox = document.querySelector('.weight')
const btn = document.querySelector('.btn-submit')
const name = document.querySelector('.input-name')
const age = document.querySelector('.input-age')
const weight = document.querySelector('.input-weight')

btn.addEventListener('click', () => {

  nameBox.forEach(nameBox => {
    nameBox.innerHTML = name.value
  })
  ageBox.innerHTML = age.value
  weightBox.innerHTML = weight.value
})
```

이벤트 타입

mouse / keyboard

class5

이벤트 타입 : 마우스 이벤트

마우스 이벤트

click : 마우스 클릭 했을때

mouseenter : 마우스 커서가 요소 위에 있을때

mouseleave : 마우스 커서가 요소 밖으로 나갈때

mouseup : 마우스 눌렀을때

mousedown : 마우스 땔때

dblclick : 더블클릭

mousemove : 마우스 이동

class5

이벤트 타입 : 마우스 이벤트

```
<div class="box"></div>  
<h1></h1>
```

```
.box {  
  width:100px;  
  height:100px;  
  background:aqua;  
}
```

class5

이벤트 타입 : 마우스 이벤트

```
const box = document.querySelector('.box')
const h1 = document.querySelector('h1')

box.addEventListener('click', () => {
  h1.innerHTML = '클릭'
})

box.addEventListener('mouseover', () => {
  h1.innerHTML = '들어옴'
})

box.addEventListener('mouseout', () => {
  h1.innerHTML = '나감'
})

box.addEventListener('mouseup', () => {
  h1.innerHTML = '뗐다'
})

box.addEventListener('mousedown', () => {
  h1.innerHTML = '누르는중'
})

box.addEventListener('dblclick', () => {
  h1.innerHTML = '더블클릭'
})
```


실습

class5

이벤트 타입 : 키보드 이벤트

키보드 이벤트

keydown : 키 눌렀을때

keyup : 키 떨어졌을때

event.대상자.key : key값 리턴

class5

이벤트 타입 : 키보드 이벤트

```
<h1>'<strong></strong>' 를 눌렀어요</h1>
```

```
<script>  
  let keyValue = document.querySelector('strong')  
  document.addEventListener('keydown', (key) => {  
    keyValue.innerText = key.key;  
  });  
</script>
```

실습

class3

과제

오늘 과제 없음