

2주차 3차시

[9장 타입 변환과 단축 평가.pdf](#)

[9장 단축 평가 \(심화\).pdf](#)

[10장 객체 리터럴.pdf](#)

9장 과제

1. 그렇다면 암묵적 타입 변환이 무조건적으로 좋지 않은 문화이자 기능인가?
 - a. 암묵적 타입 변환은 개발자가 변수를 사용할 때 마다 직접 타입을 체크할 필요를 줄여 생산성을 높여준다. 하지만 그만큼 예상치 못한 버그를 발생시킬 확률이 높아 작은 프로젝트에서는 사용하기 용이하나 큰 프로젝트거나 복잡한 코드를 사용하는 경우에는 부적절하다고 생각한다.

9장 심화 과제

1. 아래 설명에 따라 단축 평가를 이용하여 아래의 if문처럼 작동하는 true 값 여부를 판별하는 코드를 빈칸에 알맞게 작성해보시오. 결과도 내시길 바랍니다.

```
var isThereMessage = true;
var message = '';
if(isThereMessage) message = '멘토는 죽어있다.';
console.log(message);
```

```
var isThereMessage = true;
var message = '';
//if(isThereMessage) message = '멘토는 죽어있다.';
message = isThereMessage && '나도 죽어있다.';
console.log(message);
```

10장 과제

1. 아래의 코드를 실행해보시고 왜 결과 값이 그렇게 나오는지 생각해보세요

```
var person = {
  firstName : 'turtle',
  last-name : 'park'
};
console.log(person);
//=====
var word1 = {
  var: '',
  function: ''
};
console.log(word1);
//=====
//프로퍼티 키 동적 생성
var objES5 = {}
var keyES5 = 'ES5'
objES5[keyES5] = 'world';
console.log(objES5);
//=====
//계산된 프로퍼티 이름
```

```

var keyES6 = 'HELL';
var objES6 = {[keyES6]: 'o'};
console.log(objES6);
//=====
var emptyObj = {
  '' : ''
};
console.log(emptyObj);
//=====
var numObj = {
  1 : 0,
  2 : 1,
  3 : 2
};
console.log(numObj);
//=====
var duplicateObj = {
  name : 'park',
  name : 'kim'
};
console.log(duplicateObj);
//=====

```

```

{firstName: 'turtle', last_name: 'park'}
{var: '', function: ''}
{ES5: 'world'}
{HELL: 'o'}
{"": ''}
{1: 0, 2: 1, 3: 2}
{name: 'kim'}

```

추가과제1

```

var person =
{
  health: 10,
  speed: 5,
  atkdmg: 3
}

var monster =
{
  name: 'oak',
  health: 7,
  speed: 2,
  atkdmg: 5
}

while(true)
{
  if(person.speed >= monster.speed)
  {
    monster.health -= person.atkdmg;
    console.log('player hit ' + monster.name + ' (damage: ' + person.atkdmg + ') left monster health: ' + monster.health);
    if(monster.health <= 0)
    {
      console.log('person win');
      break;
    }
    person.health -= monster.atkdmg;
    console.log('monster hit player (damage: ' + monster.atkdmg + ') left person health: ' + person.health);
    if(monster.health <= 0)
    {
      console.log(monster.name + ' win');
      break;
    }
  }
  else
  {
    person.health -= monster.atkdmg;
    console.log('monster hit player (damage: ' + monster.atkdmg + ') left person health: ' + person.health);
    if(monster.health <= 0)
    {

```

```

        console.log(monster.name + ' win');
        break;
    }
    monster.health -= person.atkdmg;
    console.log('player hit ' + monster.name + ' (damage: ' + person.atkdmg + ') left monster health: ' + monster.health);
    if(monster.health <= 0)
    {
        console.log('person win');
        break;
    }
}
}
}

```

추가과제2

```

var wind = {
    'lastName' : 'park',
    1: 10
};

```

```

wind.'lastName';
wind.lastName;
wind[lastName];
wind['lastName'];
wind.1;
wind.'1';
wind[1];
wind['1']

```

```

/home/L8fUjF/prog.js:10
wind.'lastName';
      ^^^^^^^^^^^

```

```

SyntaxError: Unexpected string
    at Module._compile (internal/modules/cjs/loader.js:743:23)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:810:10)
    at Module.load (internal/modules/cjs/loader.js:666:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:606:12)
    at Function.Module._load (internal/modules/cjs/loader.js:598:3)
    at Function.Module.runMain (internal/modules/cjs/loader.js:862:12)
    at internal/main/run_main_module.js:21:11

```