

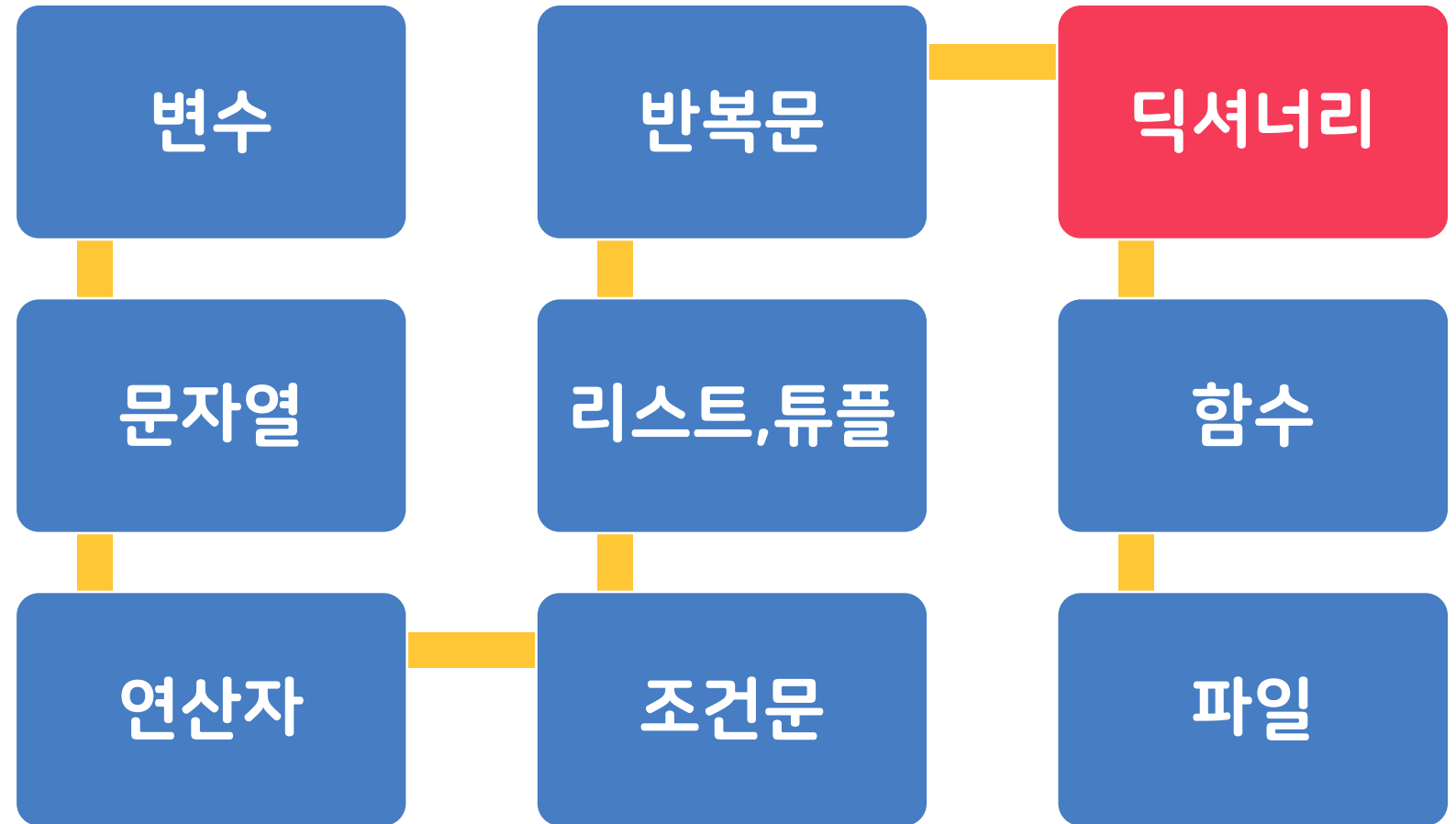


Smart Media  
스마트미디어인재개발원

임 보 미 강사



## 수업 진행방향





## 학습목표

- 딕셔너리에 대해 알 수 있다.
- 딕셔너리 함수를 활용 할 수 있다.

- dictionary는 단어 그대로 해석하면 사전이라는 뜻
- “people”이라는 단어는 “사람”, “baseball”이라는 단어는 “야구”에 부합되듯이 dictionary는 Key와 Value를 한쌍으로 갖는 자료형
- 딕셔너리 타입은 immutable한 key와 mutable한 value로 매핑되어 있는 순서가 없는 집합
- 딕셔너리 타입은 연관된 값을 묶어서 저장할 수 있는 집합  
Ex) 영표쌤반 = [상제, 현빈, 규환, 기웅]  
영표쌤반 = {'QR완료' : '상제,현빈', 'QR미완료' : '규환,기웅'}

## 1 딕셔너리(dictionary)

- 단어 그대로 해석하면 사전
- 딕셔너리는 변할 수 없는 Key와 변할 수 있는 Value를 한 쌍으로 가지고 있는 자료형이다.
- Key는 불변, Value는 가변적이다.
- 각각의 개성을 표현하기 힘들 때 딕셔너리를 사용한다
- 기호는 {}로 표현하며 구분기호는 콤마(,)를 사용한다
- Key와 Value를 한쌍으로 표현할 때 콜론(:)을 사용한다

딕셔너리명 = {Key : Value, Key : Value, ... }

※ Key에는 변하지 않는 값을 사용하고,  
Value에는 변하는 값과 변하지 않는 값 모두 사용할 수 있다.

```
a = {}
```

```
b = { "name" : "BM" }
```

```
c = { 1 : 5, 2 : 3 }
```

```
bomi = {'name': '보미', 'age': 20, 'number': '010-1234-5678'}  
bomi
```

```
{'name': '보미', 'age': 20, 'number': '010-1234-5678'}
```

```
#자료구조의 형태를 파악할 수 있는 기능  
type(bomi)
```

```
dict
```

key	value
name	보미
age	20
number	010-1234-5678

딕셔너리 정보

## 딕셔너리명[key] = value

```
bomi = {'name': '보미', 'age': 20, 'number': '010-1234-5678'}  
bomi
```

```
{'name': '보미', 'age': 20, 'number': '010-1234-5678'}
```

```
bomi['birth'] = '12/25'
```

key	value
name	보미
age	20
number	010-1234-5678
birth	12/25

딕셔너리 정보

```
bomi
```

```
{'name': '보미', 'age': 20, 'number': '010-1234-5678', 'birth': '12/25'}
```



## 딕셔너리명[key]

```
bomi
```

```
{'name': '보미', 'age': 20, 'number': '010-1234-5678', 'birth': '12/25'}
```

```
bomi['name']
```

```
'보미'
```

```
bomi['birth']
```

```
'12/25'
```

key	value
name	보미
age	20
number	010-1234-5678
birth	12/25

딕셔너리 정보

## 딕셔너리명.get(Key)

```
bomi
```

```
{'name': '보미', 'age': 20, 'number': '010-1234-5678',  
'birth': '12/25'}
```

```
bomi.get('name')
```

```
'보미'
```

```
bomi.get('birth')
```

```
'12/25'
```

key	value
name	보미
age	20
number	010-1234-5678
birth	12/25

딕셔너리 정보

## 딕셔너리명[key] VS 딕셔너리명.get(Key)

```
bomi['성별']
```

-----  
-----  
**KeyError**

Traceback

(most recent call last)

<ipython-input-55-443afa30d064> in <module>

----> 1 bomi['성별']

**KeyError:** '성별'

```
bomi.get('성별')
```

## 딕셔너리명.keys()

```
bomi = { 'name': '보미', 'age': 20, 'number': '010-1234-5678', 'birth': '12/25' }
```

```
bomi.keys()
```

```
dict_keys(['name', 'age', 'number', 'birth'])
```

```
list(bomi.keys())
```

```
['name', 'age', 'number', 'birth']
```

key	value
name	보미
age	20
number	010-1234-5678
birth	12/25

딕셔너리 정보

## 딕셔너리명.values()

```
bomi = { 'name': '보미', 'age': 20, 'number': '010-1234-5678', 'birth': '12/25' }
```

```
bomi.values()
```

```
dict_values(['보미', 20, '010-1234-5678', '12/25'])
```

```
list(bomi.values())
```

```
['보미', 20, '010-1234-5678', '12/25']
```

key	value
name	보미
age	20
number	010-1234-5678
birth	12/25

딕셔너리 정보

## 딕셔너리 for문 활용

```
for key in bomi.keys():  
    print(key)
```

name  
number  
birth

```
for value in bomi.values():  
    print(value)
```

보미  
010-1234-5678  
12/25

```
for key,value in bomi.items():  
    print(key,value)
```

name 보미  
number 010-1234-5678  
birth 12/25

## key in 딕셔너리명

- in은 딕셔너리의 키에 한에서 동작한다.

```
'name' in bom i
```

True

```
'phone' in bom i
```

False

```
'number' in bom i
```

True

key	value
name	보미
age	20
number	010-1234-5678
birth	12/25

딕셔너리 정보

## del 딕셔너리명[key]

```
bomi = {'name': '보미', 'age': 20, 'number': '010-1234-5678', 'birth': '12/25'}
```

```
bomi
```

```
{'name': '보미', 'number': '010-1234-5678', 'birth': '12/25'}
```

```
del bomi['age']
```

```
bomi
```

```
{'name': '보미', 'number': '010-1234-5678', 'birth': '12/25'}
```



## 딕셔너리명.clear()

```
bomi = { 'name' : '보미' , 'age' : 20 , 'number' : '010-1234-5678' , 'birth' : '12/25' }
```

```
bomi.clear()
```

```
bomi
```

```
{}
```

여러 학생의 성적이 Dictionary로 아래와 같이 구성되어있다.  
과목 별 합을 구하여 새로운 Dictionary로 구성하시오

```
score_dic = {'홍길동':{'수학':99,'영어':80,'국어':75},  
             '김미정':{'영어':67,'국어':87,'수학':55},  
             '정수진':{'수학':67,'국어':41,'영어':23}}
```



```
{'수학':221, '국어':203, '영어':170}
```



다음시간에는?

함수