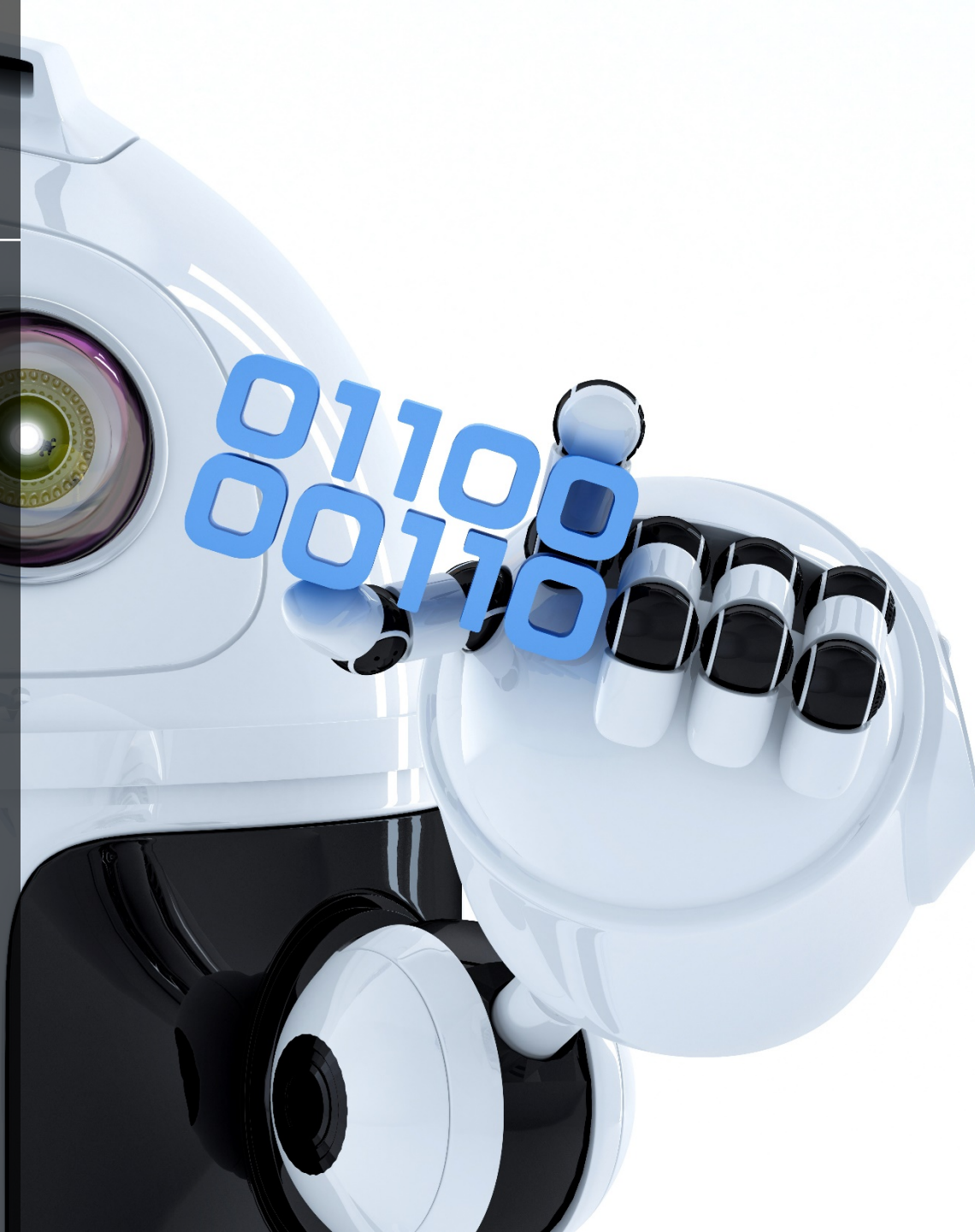


Console In & Out

Python Basic II

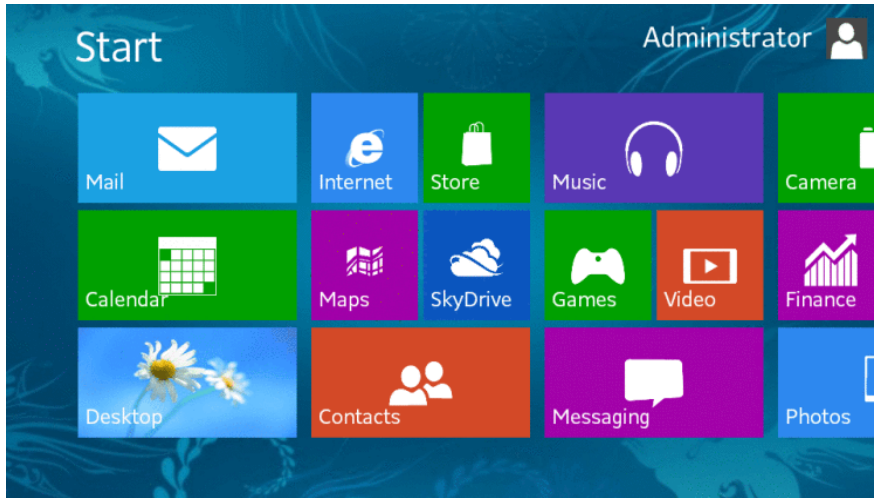
최성철 교수
Director of TEAMLAB

01100
00110

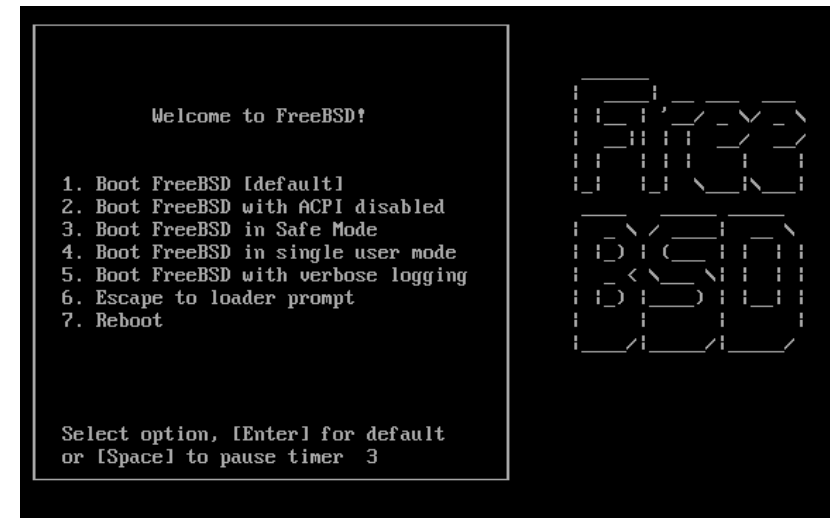


**어떻게 프로그램과
데이터를
주고 받을 것인가?**

프로그램과 소통하는 방법



Graphical **U**ser **I**nterface



Command **L**ine **I**nterface

Command Line Interface

Graphic User Interface (GUI)와 달리

Text를 사용하여 컴퓨터에 명령을 입력하는
인터페이스 체계

Windows – CMD window

Mac, Linux – Terminal

Console = Terminal = CMD창

어원: 디스플레이와 키보드가 조합된 장치

현재: CLI로 입력하는 화면



IBM 3270 terminal



IBM 7094, a typical Mainframe

Source: <http://www.gliderwiki.org/wiki/116>

콘솔창 입출력 I

input() 함수는 콘솔창에서 문자열을 입력 받는 함수

console_test.py

```
print ("Enter your name:")  
somebody = input()           # 콘솔창에서 입력한 값을 somebody에 저장  
print ("Hi", somebody, "How are you today?")
```

Editor

실행

```
python console_test.py  
Enter your name:  
cs50  
Hi cs50 How are you today?  
LoginID@cs50:~$
```

Terminal

콘솔창 입출력 II

콤마(,) 사용할 경우 print 문이 연결됨

```
>>> print ("Hello World!", "Hello Again!!!")
```

```
Hello World! Hello Again!!!
```

, 사용

실행 시 두 문장이 연결 돼서 출력됨

Python Shell

숫자 입력 받기

```
temperature = float(input("온도를 입력하세요 :"))
```

입력 시 바로 형 변환 하기

```
print(temperature)
```

temperature.py

Editor

```
python temperature.py  
온도를 입력하세요 : 103  
103.0
```

Terminal

실행



Human knowledge belongs to the world.

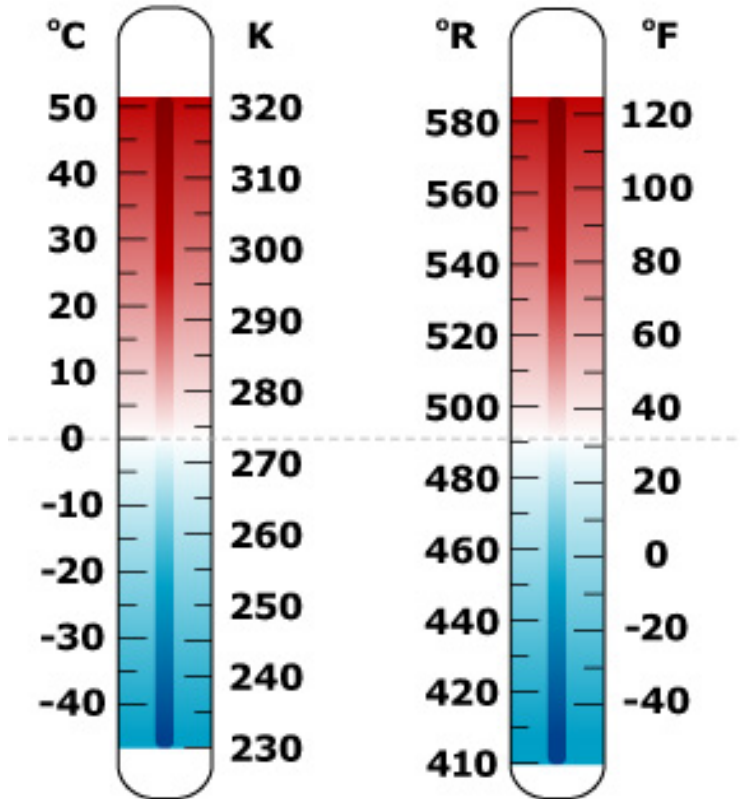
Lab: Fahrenheit Converter

Python Basic II

가천대학교 | 산업경영공학과
최성철 교수

01100
00110

섭씨와 화씨



$$^{\circ}\text{F} = (^{\circ}\text{C} \times 1.8) + 32$$

<http://goo.gl/dGQPES>

Lab: 화씨 변환기

아래와 같이 출력되는 프로그램을 만드시오

```
python fahrenheit.py
```

본 프로그램은 섭씨를 화씨로 변환해주는 프로그램입니다

변환하고 싶은 섭씨 온도를 입력해 주세요:

32.2

섭씨온도 : 32.2

화씨온도 : 89.96

Terminal

fahrenheit.py

섭씨 온도 변환 공식은: $((9/5) * \text{섭씨온도}) + 32$



Human knowledge belongs to the world.

print formatting

Python Basic II

가천대학교 | 산업경영공학과
최성철 교수

01100
00110



**형식(format)에 맞춰서
출력을 할 때가 있음**

	A	B	C	D
1	Date	Daily Income	Daily Expenses	Percent Gain/Loss
2	1-May	\$322.00	\$146.00	221%
3	2-May	\$371.00	\$135.00	275%
4	3-May	\$345.00	\$467.00	74%
5	4-May	\$345.00	\$216.00	160%
6	5-May	\$150.00	\$269.00	56%
7	6-May	\$116.00	\$481.00	24%
8	7-May	\$440.00	\$203.00	217%

**print 문을 활용해서
결과 formatting 하기**

print formatting

프린트 문은 기본적인 출력 외에 출력의 양식을 형식을 지정 가능

```
print(1, 2, 3)
print("a" + " " + "b" + " " + "c")
print("%d %d %d" % (1, 2, 3))
print("{} {} {}".format("a", "b", "c"))
```

Editor

Two types

일반적으로 %-format 과 str.format() 함수를 사용함

```
print(' %s %s' % ('one', 'two'))
```

```
print('{} {}'.format('one', 'two'))
```

```
print(' %d %d' % (1, 2))
```

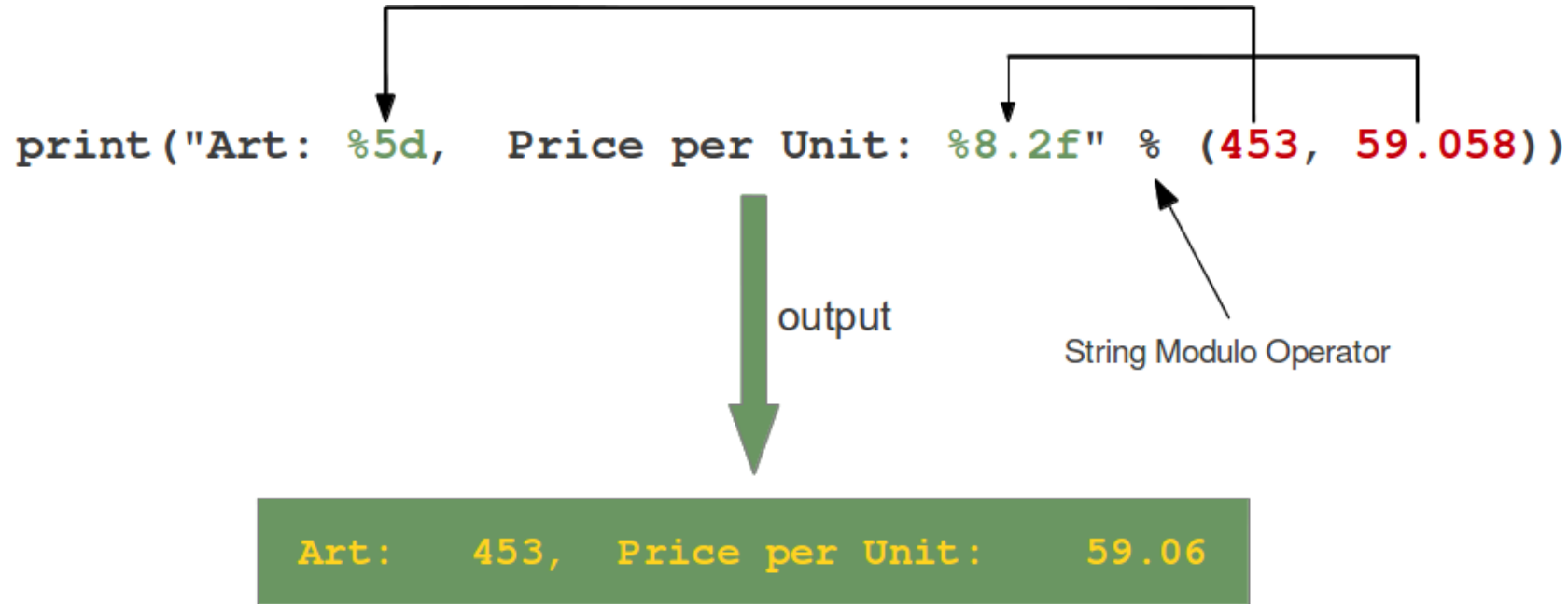
```
print('{} {}'.format(1, 2))
```

%-format

"%datatype" % (variable) 형태로 출력 양식을 표현

```
print("I eat %d apples." % 3)
print("I eat %s apples." % "five")
number = 3; day="three"
print("I ate %d apples. I was sick for %s days."
      % (number, day))
print("Product: %s, Price per unit: %f." % ("Apple", 5.243))
```

%-format



type	설명
%s	문자열 (String)
%c	문자 1개(character)
%d	정수 (Integer)
%f	부동소수 (floating-point)
%o	8진수
%x	16진수
%%	Literal % (문자 % 자체)

https://www.python-course.eu/python3_formatted_output.php

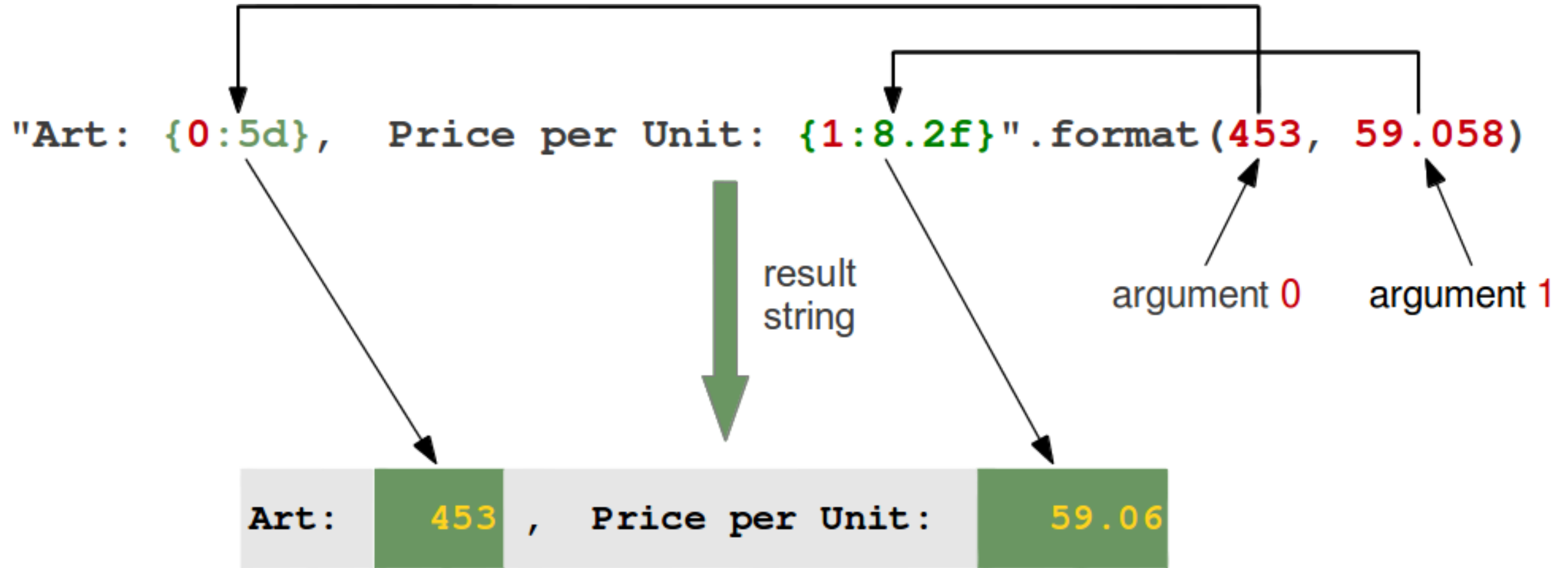
<https://wikidocs.net/13>

str.format()

"~~~~~{datatype}~~~~~".format(argument)

```
age = 36; name='Sungchul Choi'
print("I'm {0} years old.".format(age))
print("My name is {0} and {1} years old.".format(name,age))
print(
    "Product: {0}, Price per unit: {1:.3f}.".format(
        "Apple", 5.243))
```

str.format()



https://www.python-course.eu/python3_formatted_output.php

padding

여유 공간을 지정하여 글자배열 + 소수점 자릿수를 맞추기

```
print("Product: %5s, Price per unit: %.5f." % ("Apple",  
5.243))  
print("Product: {0:5s}, Price per unit:  
{1:.5f}.".format("Apple", 5.243))  
print("Product: %10s, Price per unit: %10.3f." % ("Apple",  
5.243))  
print("Product: {0:>10s}, Price per unit:  
{1:10.3f}.".format("Apple", 5.243))
```

naming

해당 표시할 내용을 변수로 표시하여 입력

```
print("Product: %(name)10s, Price per unit: %(price)10.5f." %  
      {"name": "Apple", "price": 5.243})
```

```
print("Product: {name:>10s}, Price per unit:  
{price:10.5f}.".format(name="Apple", price=5.243))
```


See

<https://docs.python.org/3/tutorial/inputoutput.html>

https://www.python-course.eu/python3_formatted_output.php

<https://wikidocs.net/13>



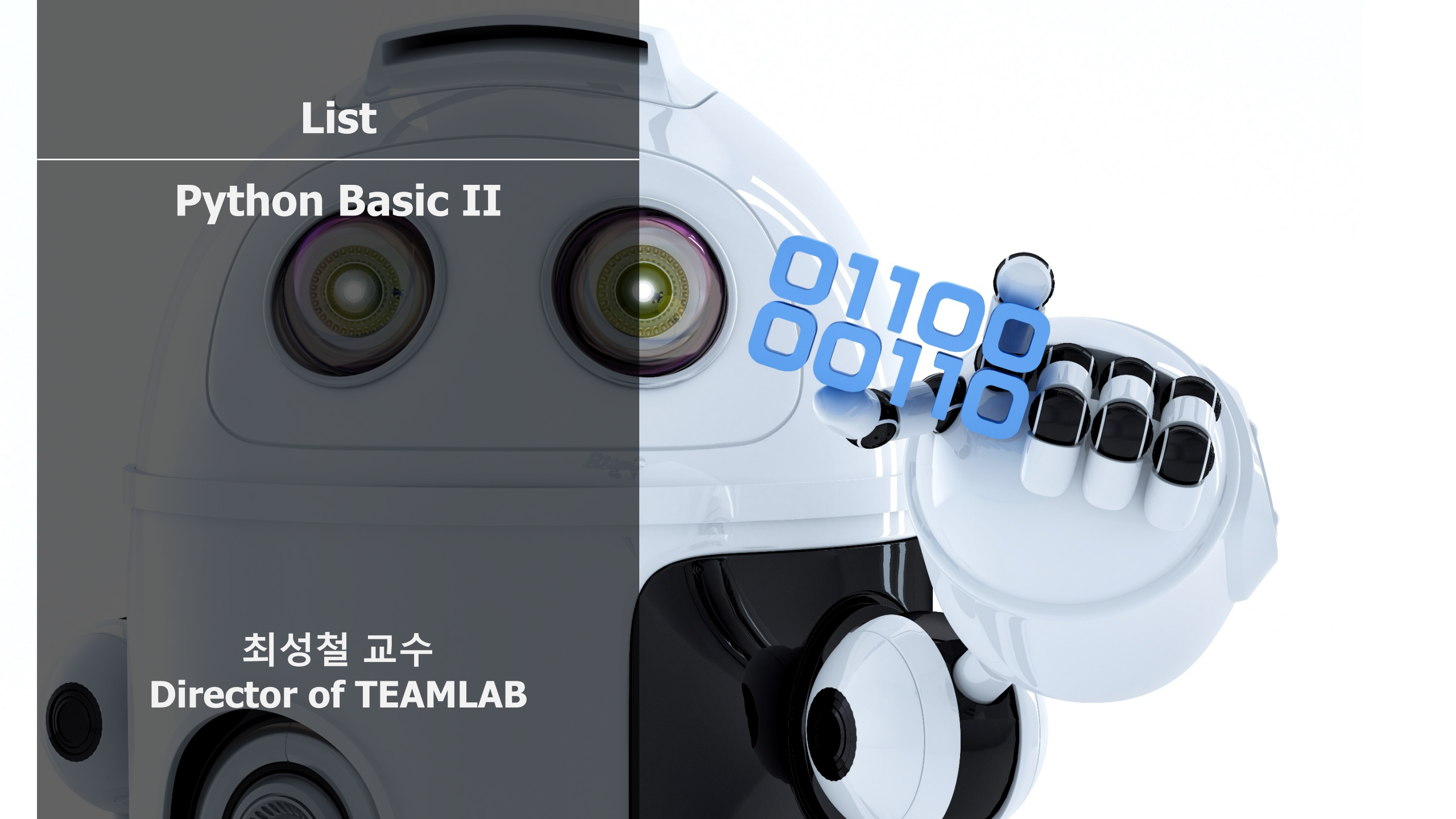
Human knowledge belongs to the world.

List

Python Basic II

최성철 교수
Director of TEAMLAB

01100
00110



데이터가 100개 있다면
어떻게 관리할 것인가?

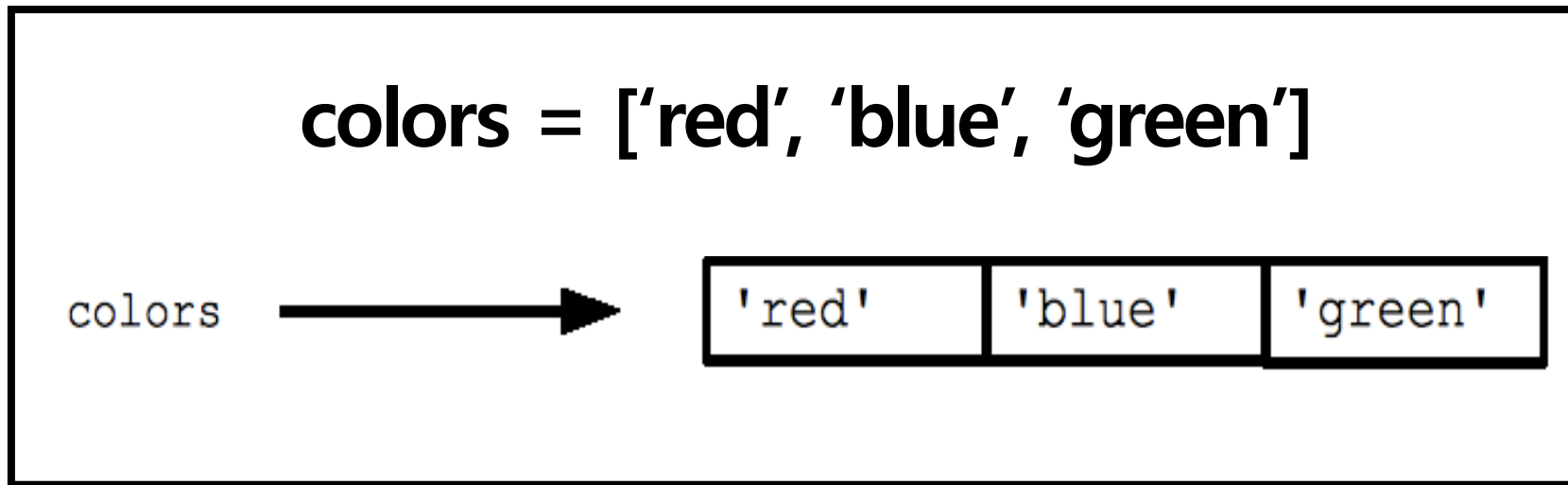
**100명의 성적 관리를
위한 변수는 몇 개?**

100개?

1개?

List 또는 Array

- 시퀀스 자료형, 여러 데이터들의 집합
- Int, Float 같은 다양한 데이터 Type 포함



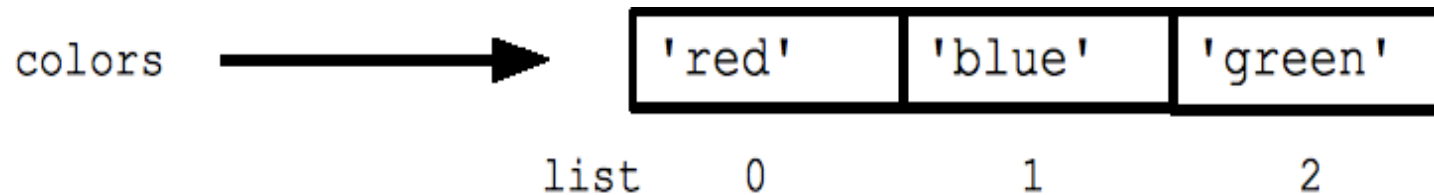
Source: <http://goo.gl/q4VvB1>, <http://goo.gl/JMbHm0>

인덱싱 (Indexing)

list에 있는 값들은 주소(offset)를 가짐,
주소를 사용해 할당된 값을 호출

```
colors = ['red', 'blue', 'green']  
print (colors[0])    # red  
print (colors[2])    # green  
print (len(colors))  # 3  
                    # len은 list의 길이를 반환
```

Editor



Source: <http://goo.gl/q4VvB1>, <http://goo.gl/JMbHm0>

슬라이싱 (Slicing)

- list의 값들을 잘라서 쓰는 것이 슬라이싱
- list의 주소 값을 기반으로 부분 값을 반환

Editor

```
cities = ['서울', '부산', '인천', '대구', '대전', '광주', '울산', '수원']  
print (cities[0:6], " AND ", a[-9:]) # a 변수의 0부터 5까지, -9부터 끝까지  
print (cities[:])    # a변수의 처음부터 끝까지  
print (cities[-50:50]) # 범위를 넘어갈 경우 자동으로 최대 범위를 지정  
print (cities[::2], " AND ", a[:: -1]) # 2칸 단위로, 역으로 슬라이싱
```


리스트의 연산

- 인덱싱, 슬라이싱, 연산 등 활용

```
>>> color = ['red', 'blue', 'green']
>>> color2 = ['orange', 'black', 'white']
>>> print(color + color2)    # 두 리스트 합치기
>>> len(color) # 리스트 길이
>>> color[0] = 'yellow' # 0번째 리스트의 값을 변경
>>> print(color * 2)    # color 리스트 2회 반복
>>> 'blue' in color2    # 문자열 'blue'가 color2 존재 여부 반환
>>> total_color = color + color2
```

Python Shell

리스트 추가와 삭제

- append, extend, insert, remove, del 등 활용

이전 장과 연결 돼서 실행

Python Shell

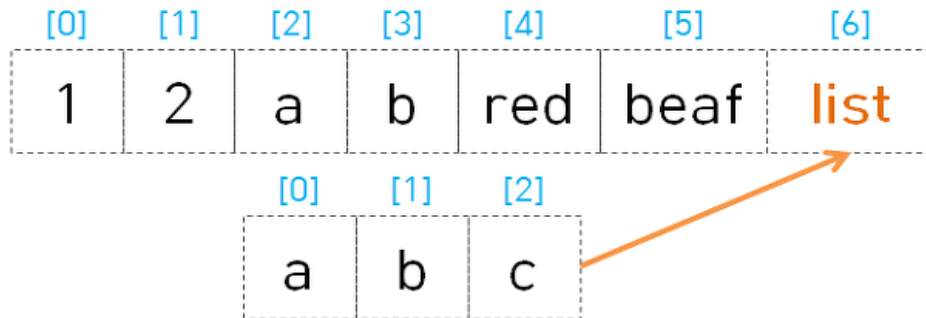
```
>>> color.append("white")      # 리스트에 "white" 추가
>>> color.extend(["black","purple"]) # 리스트에 새로운 리스트 추가
>>> color.insert(0,"orange")    # 0번째 주소에 "orange" 추가
>>> print (color)
['orange', 'yellow', 'blue', 'green', 'white', 'black', 'purple']
>>> color.remove("white")      # 리스트에 "white" 삭제
>>> del color[0]               # 0번째 주소 리스트 객체 삭제
>>> print (color)
['yellow', 'blue', 'green', 'black', 'purple']
```

Python 리스트만의 특징

- 다양한 Data Type이 하나에 List에 들어감

```
>>> a = ["color", 1, 0.2]
>>> color = ['yellow', 'blue', 'green', 'black', 'purple']
>>> a[0] = color # 리스트 안에 리스트도 입력 가능
print (a)
[['yellow', 'blue', 'green', 'black', 'purple'],
 1, 0.20000000000000001]
```

Python Shell



중첩 리스트 시 메모리 구조

Source: <http://goo.gl/FApwnw>

리스트 메모리 저장 방식

```
>>> a = [5, 4, 3, 2, 1]
```

Python Shell

```
>>> b = [1, 2, 3, 4, 5]
```

```
>>> b = a
```

```
>>> print (b)
```

```
[5, 4, 3, 2, 1]
```

```
>>> a.sort()
```

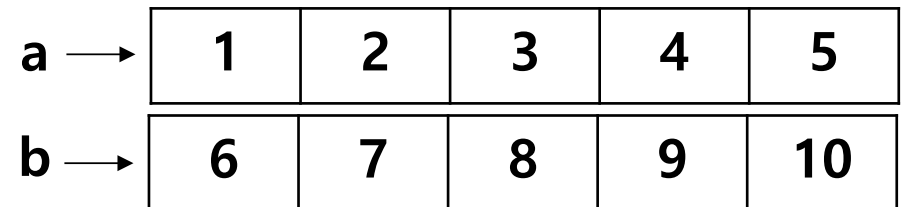
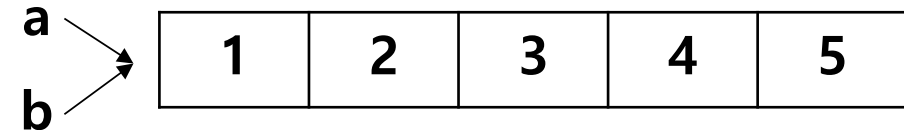
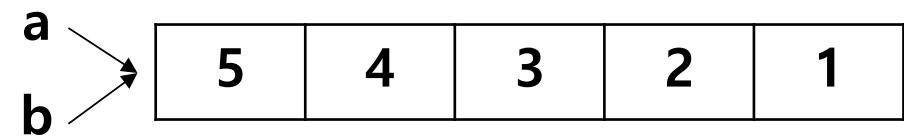
```
>>> print (b)
```

```
[1, 2, 3, 4, 5]
```

```
>>> b = [6,7,8,9,10]
```

```
>>> print (a, b)
```

```
[1, 2, 3, 4, 5] [6, 7, 8, 9, 10]
```



"=" 의 의미는 같다가 아닌 메모리 주소에 해당 값을 할당(연결)한다는 의미

패킹과 언패킹

- 패킹 : 한 변수에 여러 개의 데이터를 넣는 것
- 언패킹 : 한 변수의 데이터를 각각의 변수로 반환

```
>>> t = [1, 2, 3]           # 1,2,3을 변수 t에 패킹
>>> a, b, c = t             # t에 있는 값 1, 2, 3 을 변수 a, b, c에 언패킹
>>> print(t, a, b, c)      # [1, 2, 3] 1 2 3
[[1,2,3] 1 2 3]
```

Python Shell

이차원 리스트

- 리스트 안에 리스트를 만들어 행렬(Matrix) 생성

```
>>> kor_score = [49,79,20,100,80]
>>> math_score = [43,59,85,30, 90]
>>> eng_score = [49,79,48,60,100]
>>> midterm_score = [kor_score, math_score, eng_score]
>>> print (midterm_score[0][2])
```

20

Python Shell

	A	B	C	D	E
국어점수	49	79	20	100	80
수학점수	43	59	85	30	90
영어점수	49	79	48	60	100



Human knowledge belongs to the world.