파이썬 포트폴리오

20191772
최원기

개요

Python은 고급 프로그래밍 언어로 플랫폼에서 독립적이며 인터프리터식, 객체지향적, 동적 타이핑의 대화형 언어이다. 오픈소스로 누구나 다운받아 사용가능하다. 가독성이 좋고 작성이 다른 코드에 비해 쉽다는 것이 특징이다.

제 4차 산업혁명 시대는 모든 사물이 연결된 초연결 사회로 모든 산업은 컴퓨터 과학 공학 기술을 기반으로 융합된다. 이는 컴퓨터 사고력 없이는 문제 해결이 불가능해 지게 된다. 컴퓨팅 사고력은 그 외에 모든 일상생활에서 문제 해결능력을 키워주고 창의력 발전에 힘이 되기에 모두가 키워야 할 기본 역량이다.

파이썬을 공부하며 우리는 컴퓨팅 사고력을 키워 제 4차 산업혁명을 시대를 이끌어가자.

목차

쉘 실행

```
>>> print("helloworld")
helloworld
>>> print("hellow, world")
hellow, world
>>> print(1)
1
>>> print(1000)
1000
>>> print(잘생긴 남자)
SyntaxError: invalid syntax
>>> print(helloworld)
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    print(helloworld)
NameError: name 'helloworld' is not defined
```

Print는 쉘에 글자를 출력하는 역할을 한다

Print를 작성하고 ( ) 안에 “ ” 출력하고자 하는
글자를 입력하고 Enter를 누르면 출력된다.

숫자를 출력하고 싶을 시에는 “ ” 없이 입력하면 된다.

문자를 “ ” 감싸지 않으면 에러가 발생한다.

추가로 문자를 변수로서 특정 값을 지정해줬었다면

그 값이 출력된다.

자료형

앞서 Hello world 를 출력할 때와 1 을 출력할 때 방식이 달랐는데 이는 문자의 자료형이 다르기때문이다. 자료에는 크게 문자와 수가 있다. 문자열은 string이라 하며 따옴표로 감싸면 모두 문자로 받아들인다. 예를 들어 “1”은 숫자로 1 이 입력 된 것이 아니라 문자 1 로 받아들여진다.

```
>>> print(1234, "qwerty", "mnbvcxz")
1234 qwerty mnbvcxz
```

여기서 1234는 수로 입력 되었고 나머지는 “ ”로 감싸지며 문자로 입력 되었다. 여러 자료를 한번에 출력하는 방법은 ,를 이용하는 것이다. 이때 사이에 자동으로 공백이 추가된다.

```
>>> 'qwert' + 'yuiop'
'qwertyuiop'
>>>
```

두 문자열을 완전히 붙이려면 +를 사용하면 된다.

자료형

```
>>> '으악 학교가고싶다'*4
'으악 학교가고싶다으악 학교가고싶다으악 학교가고싶다으악 학교가고싶다'
>>>
```

연산 기호를 이용 할 수도 있는데 문자열에 * 를 이용하고
숫자를 입력해주면 그 수 만큼 출력한다.

```
>>> x = '1234567890'
>>> x[4]
'5'
>>>
```

[ ] 는 그 문자열의 순서에 맞는 문자를 출력해준다.
추가로 python은 0부터 수를 세기 때문에 [4]는 4가 아니라 5가 된다.

자료형

```
>>> 3 + 4 # 덧셈
7
>>> 3 - 4 # 뺄셈
-1
>>> 3 * 4 # 곱셈
12
>>> 3 / 4 # 나눗셈
0.75
>>> 3 // 4 # 나눗셈 : 몫 (a=bq+r일 때 q를 출력)
0
>>> 3 % 4 # 나머지 (modulo, a mod b)
3
>>> 3 ** 4 # 거듭제곱
81
>>>
```

여러가지 연산이 가능하다. 각 기호들이다.

수에도 여러가지 형식이 있는데
크게 정수(int) 실수(float) 복소수(complex) 가 있다.
복소수에서 우리가 알고 있는 i가 아닌 j로 사용한다.

```
>>> x = 70
>>> print(x)
70
>>> print(type(x))
<class 'int'>
>>>
```

```
>>> x = 36.5
>>> print(x)
36.5
>>> print(type(x))
<class 'float'>
>>>
```

```
>>> x = 2 + 3j
>>> print(x)
(2+3j)
>>> print(type(x))
<class 'complex'>
>>>
```

변수

```
>>> x = 7
>>> y = "Sean"
>>> print(x)
7
>>> print(y)
Sean
>>>
```

파이썬에서 변수 지정은 자료형에 따라 다른데 문자열은 “”를 이용해 묶고 숫자열은 그냥 = 뒤에 바로 사용하면 된다.

x 를 7 로 지정했고 y 를 “Sean” 으로 변수 지정하여 출력시
7과 Sean 이 출력 되는 것을 볼 수 있다.

```
>>> x = 7
>>> x = "Sean"
>>> print(x)
Sean
>>>
```

이렇게 여러 번 지정 해 주었을 경우 마지막에 지정된 값으로 출력된다.

x 를 7로 지정했으나 이후 “Sean” 으로 다시 지정해 결론적으로 Sean 이 출력된다.

조건문

if 는 만약에 라는 뜻으로 코딩에선 어떠한 조건을 성립 했을 시에 이벤트를 발생시키는 역할을 한다.

Else는 if조건이 성립하지 않았을 때 이벤트를 발생 시킨다.

else if 는 조건을 여러 개를 사용할 때 이용된다.

```
if x % 2 == 0:
x = str(x)
x = x + '는(은) 짝수입니다.'
print(x)
else:
x = str(x)
x = x + '는(은) 홀수입니다.'
print(x)
```

만약에 x를 2 로 나누었을 때 나머지가 0 이라면 짝수 라고 한다.

그렇지 않다면 (else) 홀수 라고 한다.

조건문

```
if x % 2 == 0:
x = str(x)
x = x + '는(은) 짝수입니다.'
print(x)
elif x % 2 == 1:
x = str(x)
x = x + '는(은) 홀수입니다.'
print(x)
```

만약에 x를 2 로 나누었을 때 나머지가 0 이라면 짝수 라고 한다.

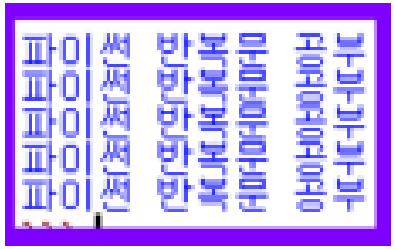
그렇지 않고(else) 2로 나누었을 때 나머지가 1 이라면(if) 홀수 라고 한다.

아까와 같은 결과를 나타내지만 else와 elif의 차이는 그렇지 않다면 의 else 에 추가적인 조건이 있냐 없냐 이다.

조건문

For 는 특정 조선 속에서 이벤트를 반복시키는 역할을 한다.

```
for i in [1,2,3,4,5]:
    print("파이썬 반복문 공부")
```



i가 1,2,3,4,5 로 지정 되어있어 print를 5번을 반복한다.

```
for i in ["아두이노","라즈베리파이","마이크로비트"]:
    print(i)
```

```
아두이노
라즈베리파이
마이크로비트
```

i를 문자열로 저장후 print하면 i의 순서대로 출력된다.

마무리

이렇게 파이썬의 기본 코딩에 대해 알아보았다. 아직은 부족한 부분도 많아 세세하게 더 채워가고 싶은 마음이 들었다. 1년 넘는 기간 동안 전공으로 컴퓨터에 대해 공부 해왔다. 여러가지 프로그램을 이용해 많은 프로젝트를 만들며 앞서 말했던 컴퓨팅 사고력을 키울 수 있었다. 현재는 이를 전반적인 모든 일상에 적용하여 문제를 해결하고 있다. 문제를 해결할 방안을 구색하는 뿐 아니라 거기서 오류가 발생하였을 때에 해결 방안까지 생각을 하는 능력은 나의 학업 능력을 더욱 증진시켰다.

나와 같은 방법으로 많은 사람들이 컴퓨팅 사고력을 키워 제 4차 산업혁명을 일으킬 수 있길 기대하며 다음 번에 기회가 된다면 이 포트폴리오를 시작으로 점차 더욱 세세한 포트폴리오를 만들어 많은 사람들에게 나의 정보를 이해 하기 쉽도록 설명하고자 하는 목표를 세웠다.