

Python 문법 기초

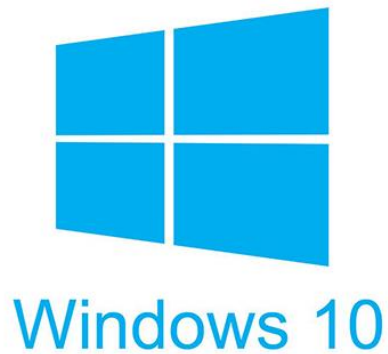
3번째 세션

NEXT X LIKELION 전해주

| 프로그램이란?

컴퓨터 안에 있는 논리적인 기계

컴퓨터를 실행시키기 위해 쓰여진 명령어들의 모음



| 프로그래밍 언어

프로그램을 작성할 때 쓰는 언어

프로그램을 작성하기 위한 형식언어



컴퓨터한테 명령을 내려서 일을 시킨다!

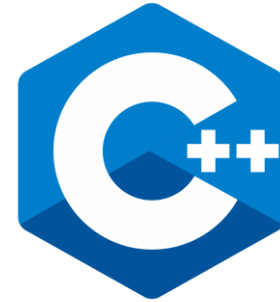
| 프로그래밍 언어

종류가 아주 많아요

인간의 언어와 더 비슷해서 쉽다

THE
C
PROGRAMMING
LANGUAGE

 PASCAL



JavaScript



PYTHON

파이썬이란

1989년 12월..

“크리스마스에 특별히 할 일이 없었다.
연구실은 닫혔고 집에 컴퓨터가 있길래
파이썬을 만들었다”

-귀도 반 로섬-



파이썬의 응용분야

웹개발, 머신러닝, 딥러닝 등등

django



PYTORCH



PYTHON



파이썬의 단점

하지만 느리고 앱 개발이 불가능하다

코드를 한 줄씩 읽기 때문에 속도가 느리고,
모바일 앱 개발은 불가능합니다.

주의할 점

파이썬에서 들여쓰기는 필수

들여쓰기 절대 잊지 마세요!!!

```
1  for i in range(10):  
2  |    print(i)
```

```
4  def add(a, b):  
5  |    return a + b
```


이제 실습을 시작해봅시다!

[터미널에서 인터프리터 실행]

윈도우는 `python -i`

맥은 `python3`

[터미널에서 인터프리터 종료]

`exit()`

변수

프로그래밍의 기초

변수란 값을 저장하기 위한 이름

변수명 = 저장할 값

등호 오른쪽 값을 왼쪽에 저장하겠다!
파이썬은 변수 type을 지정해줄 필요가 없습니다.

```
>>> a = 3
>>> a
3
>>> b = a
>>> a
3
>>> b = 4
>>> b
4
```

- 숫자형
- 문자열
 - 불
- 리스트
 - 튜플
- 딕셔너리
 - 집합

| 숫자형

사칙연산

덧셈	+
뺄셈	-
곱셈	*
나눗셈	/
제곱	**
나머지	%
몫	//

```
>>> 3 + 2
5
>>> 3 - 2
1
>>> 3 * 2
6
>>> 3 / 2
1.5
>>> 3 ** 2
9
>>> 3 % 2
1
>>> 3 // 2
1
```

문자열

쉬운 문자열 연산은 파이썬의 장점

문자열은 큰따옴표 또는 작은따옴표로 묶어줍니다.
(둘 중 아무거나 써도 상관없음)

```
>>> a = "likelion"
>>> a
'likelion'
>>> b = 'likelion'
>>> b
'likelion'
```

문자열

쉬운 문자열 연산은 파이썬의 장점

문자열 연결	문자열1 + 문자열2 + ...
문자열 반복	문자열 * 반복횟수

```
>>> a = "I like"
>>> b = "python"
>>> a + " " + b
'I like python'
```

```
>>> b * 3
'pythonpythonpython'
```

문자열

쉬운 문자열 연산은 파이썬의 장점

인덱싱 (특정 위치 문자 반환)	변수이름[인덱스 번호]
슬라이싱 (일부 문자열 추출)	변수이름[시작 인덱스, 끝 인덱스+1]

인덱스는 항상 0부터 시작합니다!

0은 첫번째 인덱스, -1은 마지막 인덱스

```
>>> c[0]
'p'
>>> c[1]
'y'
>>> c[2]
't'
>>> c[-1]
'n'
>>> c[0:2]
'py'
>>> c[0:3]
'pyt'
```

문자열 관련 함수

쉬운 문자열 연산은 파이썬의 장점

문자열 길이 계산	len(문자열)
문자 개수 세기	문자열.count(세고자 하는 문자)
문자 위치 찾기	문자열.find(찾고자 하는 문자) 있으면 처음 인덱스 없으면 -1 반환

```
>>> a = "likelion"
>>> a.count("l")
2
>>> a = "lion"
>>> len(a)
4
>>> b = "likelion"
>>> b.count("l")
2
>>> b.find("l")
0
>>> b.find("k")
2
>>> b.find("p")
-1
```


문자열 관련 함수

쉬운 문자열 연산은 파이썬의 장점

문자열 삽입	삽입할 문자.join(대상 문자열)
소문자를 대문자로	문자열.upper()
대문자를 소문자로	문자열.lower()

```
>>> a = "aaaaaaa"
>>> "?".join(a)
'a?a?a?a?a?a'
```

```
>>> b = "Likelion"
>>> b.upper()
'LIKELION'
>>> b.lower()
'likelion'
```

문자열 관련 함수

쉬운 문자열 연산은 파이썬의 장점

양쪽 공백 제거	문자열.strip()
문자열 나누기	문자열.split(기준 문자) 기준문자 안 쓰면 공백 기준 분리
문자열 바꾸기	문자열.replace(변화 전 문자열, 변화 후 문자열)

```
>>> a.strip()
'I like python'
>>> b = "a?a?a?a?a"
>>> b.split("?")
['a', 'a', 'a', 'a', 'a']
>>> b.replace("?", "")
'aaaaa'
```

문자열 실습 문제

1. 공지방에서 session3 파일을 받아 압축을 풀어주세요
 2. string.py에 있는 문제를 풀어주세요!
- * python 파일명.py 를 입력하면 파일이 실행됩니다

정답

```
4  answer = "3/29은 NEXT 세션하는 날!"
5
6  string1 = "2021-03-29은 "
7  string2 = "next"
8  string3 = "      세션하는 날!      "
9
10 string1 = string1.replace("2021-0", "").replace("-", "/")
11 string2 = string2.upper()
12 string3 = string3.strip()
13
14 string = string1 + string2 + " " + string3
15
16 if answer == string:
17     print("정답입니다!")
18 else:
19     print("틀렸습니다.")
20     print(f"정답 : {answer}")
21     print(f'입력값 : {string}')
```

| Bool 자료형

참 또는 거짓!

값 대입	=
비교문자(같다)	==
비교문자(다르다)	!=

```
>>> 2 == 2
True
>>> 2 == 3
False
>>> 2 != 3
True
```

리스트 자료형

다른 언어의 array, 자유롭게 수정이 가능하다

리스트 연결	리스트1 + 리스트2 + ...
리스트 반복	리스트 * 반복횟수
인덱싱 (특정 위치 문자 반환)	리스트이름[인덱스 번호]
슬라이싱 (일부 문자열 추출)	리스트이름[시작 인덱스, 끝 인덱스+1]

```
>>> a = [1,2,3,4]
>>> b = ["1","2","3","4"]
>>> c = []
>>> d = list()
```

외울 필요 없어요! 문자열과 똑같습니다

리스트 자료형

다른 언어의 array, 자유롭게 수정이 가능하다

맨 뒤에 삽입	리스트.append(삽입할 요소)
특정 위치 삽입	리스트.insert(인덱스, 삽입할 값)
맨 뒤에 삭제	리스트.pop()
특정 위치 삭제	del 리스트[인덱스] 리스트.remove(아이템)

기타 등등 함수들은 구글링으로 더 알아보세요

```
>>> a = [1,2,3]
>>> a.append(4)
>>> a
[1, 2, 3, 4]
>>> a.pop()
4
>>> a
[1, 2, 3]
>>> a.insert(3,4)
>>> a
[1, 2, 3, 4]
>>> del a[3]
>>> a
[1, 2, 3]
```

| 튜플 자료형

잘 안 써요..

리스트와 유사하지만 수정, 삽입, 삭제가 불가능합니다.

수정되어서는 안될 내용을 정의할 때 주로 사용해요!

```
>>> a = (1,2,3,4)
>>> b = tuple()
>>> a[0]
1
>>> a.append(5)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'tuple' object has no attribute 'append'
```


| 딕셔너리 자료형

말 그대로 사전!

Key값으로 접근하여 Value값을 가져올 수 있습니다.

```
>>> user_dict = {  
...     "name" : "haeju",  
...     "age" : "24",  
...     "college" : "engineering"  
... }  
>>> user_dict["name"]  
'haeju'  
>>> user_dict["age"]  
'24'
```

| Set 자료형

이것도 말 그대로 집합!

중복 제거에 주로 사용됩니다.

집합이어서 순서는 보존되지 않아요

```
>>> user_set = set(["1", "1", "2", "3", "3", "4"])  
>>> user_set  
{'1', '4', '3', '2'}
```

조건문

들여쓰기는 필수!

```
16 money = 200
17
18 if money > 5000:
19     print("택시를 타고 가세요")
20 elif money > 1250:
21     print("지하철을 타고 가세요")
22 else:
23     print("걸어가세요")
```

| 포함여부

In 또는 not in

```
26   likelion = ["단비", "연우", "동현", "지현"]
27
28   if "해주" in likelion:
29       |   print("멋사 소속입니다.")
30   elif "해주" not in likelion:
31       |   print("멋사 소속이 아닙니다.")
```

조건문 실습 문제

1. 공지방에서 session3 파일을 받아 압축을 풀어주세요
 2. control.py에 있는 문제를 풀어주세요!
- * python 파일명.py 를 입력하면 파일이 실행됩니다

정답

```
3 list1 = ["갈비탕", "비빔밥", "냉면"]
4 list2 = ["짜장", "짬뽕", "꿔바로우", "깐풍기"]
5 list3 = ["파스타", "스테이크", "감바스"]
6
7 food = input("음식을 입력해주세요: ")
8
9 # 여기서부터 음식이 한식, 중식, 양식 중 어느 곳에 속하는지
판단하는 코드를 작성해주세요. 단, 조건식으로 list1, list2,
list3에 포함되는지 여부를 이용하셔야 합니다!
10
11 if food in list1:
12     print("한식입니다.")
13 elif food in list2:
14     print("중식입니다.")
15 else:
16     print("양식입니다.")
```

빠져나갈 조건이 필요합니다!

```
1   count = 1
2
3   ✓ while count < 5:
4       |   print("안녕하세요")
```

빠져나갈 조건이 필요합니다!

```
1  count = 1
2
3  ✓ while count < 5:
4      print("안녕하세요")
5      count = count + 1
```

```
5  count = 1
6
7  ✓ while count < 5:
8      print("%d번째 반복 중" % count)
9      count = count + 1
```


range(시작 인덱스, 끝 인덱스 + 1, step)

```
1  for i in range(1, 6, 1):  
2      print("%d번째 반복 중" % i)  
3      print("안녕하세요")
```

```
1  for i in range(5):  
2      print("%d번째 반복 중" % i)  
3      print("안녕하세요")
```

반복문

For / while

```
1  count = 1
2
3  while count < 5:
4      if count == 3: # count가 3이 되면 반복문 탈출
5          break
6
7      print("%d번째 반복 중" % count)
8      count = count + 1 # count += 1
9
10 for i in range(5):
11     if i % 2 == 0: # i가 짝수라면
12         continue # 이번 반복에서는 내 밑 코드는 전부 skip
13
14     print("%d번째 반복 중" % i)
```

반복문 실습 문제

1. 공지방에서 session3 파일을 받아 압축을 풀어주세요
 2. loop1.py와 loop2.py에 있는 문제를 풀어주세요!
(시간이 남으신 분들은 추가 문제도 풀어보세요!)
- * python 파일명.py 를 입력하면 파일이 실행됩니다

정답

```
1  # 문제1. for문을 활용해서 1부터 10까지 숫자를 출력해주세요.
2
3  for i in range(1, 11, 1):
4      print(i)
5
6  # 문제2. for문을 활용해서 1부터 10까지 숫자 중 3의 배수는
   제외하고 출력해주세요. (단, continue이용)
7
8  for i in range(1, 11, 1):
9      if i % 3 == 0:
10         continue
11     print(i)
```

정답

```
1  # while문을 사용하여 아래와 같이 별을 표시하는 프로그램을  
   작성해주세요.  
2  
3  ""  
4  *  
5  **  
6  ***  
7  ****  
8  *****  
9  ""  
10  
11  i = 1  
12  
13  while i < 6:  
14      print("*" * i)  
15      i = i + 1
```

정답

```
24  """
25  *****
26  |  ****
27  |    ***
28  |     **
29  |      *
30  """
31  # 풀0/1
32  for i in range(5):
33      for j in range(5):
34          if j < i:
35              print(" ", end = "")
36          else:
37              print("*", end = "")
38      print()
39
40  # 풀0/2
41  for i in range(5, 0, -1):
42      print(" " * (5 - i), end = "")
43      print("*" * i)
```

포매팅

들여쓰기는 필수!

정수	%d
실수	%f
문자	%c
문자열	%s

Enter	\n
Tab	\t

```
26  # %포매팅
27  print("정수 출력 %d" % 123)
28  print("실수 출력 %f" % 123.12)
29  print("문자 출력 %c \t 문자열 출력 %s" % ("호", "안녕"))
30
31  # format 포매팅
32  name = "전해주"
33  print("제 이름은 {}입니다.\n", name)
34
35  # f-string
36  name = "전해주"
37  print(f'제 이름은 {name}입니다.')
```

함수란

Input을 넣으면 output이 나온다

특정 작업을 여러 번 반복해야 할 때

여러 번 같은 코드를 적는 대신

1. 해당 작업을 수행하는 함수를 작성하고
2. 필요할 때마다 함수를 호출합니다!

함수란

return이 두 번 나올 수는 없어요

코드를 다 실행하거나

return을 만나면 함수가 종료됩니다.

따라서 함수의 결과값은 최대 한 개입니다!

```
44  # 입력값 0, 결과값 0
45  ✓ def add(a, b):
46      |     return a + b
47
48  # 입력값 0, 결과값 x
49  ✓ def show_add(a, b):
50      |     print(a + b)
51
52  # 입력값 x, 결과값 d
53  ✓ def get_number():
54      |     return 4
55
56  # 입력값 x, 결과값 x
57  ✓ def say_hello():
58      |     print("hello")
```

함수란

return이 두 번 나올 수는 없어요

함수를 먼저 정의해야 함수를 실행할 수 있어요!

함수이름(인자)

```
61  ✓ def add(a, b):  
62      |     return a + b  
63  
64  print("1과 2를 더하면 결과는 %d입니다.", add(1,2))
```

함수 실습 문제

1. 공지방에서 session3 파일을 받아 압축을 풀어주세요
 2. def.py에 있는 문제를 풀어주세요!
- * python 파일명.py 를 입력하면 파일이 실행됩니다

정답

```
1  #입력받은 문자열의 역순을 출력하는 함수를 작성하세요.  
2  #ex) python -> nohtyp  
3  
4  string = input("문자열을 입력해주세요: ")  
5  
6  def reverse(str):  
7      for i in range(len(str)-1, -1, -1):  
8          print(str[i], end = "")  
9  
10 reverse(string)
```

Session3 과제1

hw1.py와 hw2.py에 있는 문제를 풀어주세요

(인터넷에 정답 검색은 하지 말고 스스로 해보시는 것을 추천드려요)

마감기한 : 4월 1일까지

Session3 과제2

페이스북 클론코딩

마감기한 : 4월 5일까지

예시

