파이썬과 친해지기

뱉는다(반환)의 개념, 10주차 복습, 세계 속 파이썬

12주차

“뱉는다”?

input() 함수

사용자 입력

input() 함수는 사용자의 입력을 “뱉는다”.

“뱉는다”?

input() 함수는 **사용자의 입력**을 뱉는다.

1 + 1 은 2를 뱉는다.

“안녕하세요?”[1] 은 **“녕”**를 뱉는다.

“뱉는다”라는 개념은 매우 중요!

뱉는다의 개념 - 예제 1

```
a = "치킨"
b = "피자"
print(a + b)
```

이 줄이 어떻게 바뀌는지 보자.

뱉는다의 개념 - 예제 1

```
a = "치킨"
b = "피자"
print("치킨" + b)
```

이 줄이 어떻게 바뀌는지 보자.
a 는 “치킨”을 뱉는다.

뱉는다의 개념 - 예제 1

```
a = "치킨"
b = "피자"
print("치킨" + "피자")
```

이 줄이 어떻게 바뀌는지 보자.
a 는 “치킨”을 뱉는다.
b 는 “피자”를 뱉는다.

뱉는다의 개념 - 예제 1

```
a = "치킨"
b = "피자"
print("치킨피자")
```

이 줄이 어떻게 바뀌는지 보자.
a 는 “치킨”을 뱉는다.
b 는 “피자”를 뱉는다.
“치킨” + “피자” 는 “치킨피자”를 뱉는다.

뱉는다의 개념 - 예제 1

```
a = "치킨"
b = "피자"
print(a + b)
```

print() 함수 안에 쓴 것은 a + b 이지만
실제로 함수에게 전달된 것은 “치킨피자”이다.

뱉는다의 개념 - 예제 1

```
a = "치킨"
b = "피자"
print(a + b)
```

print() 함수 안에 쓴 것은 a + b 이지만
실제로 함수에게 전달된 것은 “치킨피자”이다.

```
>>> %Run hey.py
치킨피자
```

즉 print() 함수는 “a + b”가 아닌 “치킨피자”를 출력.

뱉는다의 개념 - 예제 2

```
a = input("이름을 입력하세요.")
print("당신의 이름은", a, "입니다.")
```

```
>>> %Run hello.py
 이름을 입력하세요.
```

뱉는다의 개념 - 예제 2

```
a = "우주최강염소"
print("당신의 이름은", a, "입니다.")
```

```
>>> %Run hello.py
 이름을 입력하세요.우주최강염소
```

입력 후 엔터를 치는 순간 input() 이 “우주최강염소” 를 뱉는다.

받는다의 개념 - 예제 2

```
a = "우주최강염소"
print("당신의 이름은", "우주최강염소",
```


```
>>> %Run hello.py
 이름을 입력하세요.우주최강염소
```

입력 후 엔터를 치는 순간 input() 이 “우주최강염소” 를 받는다.
a는 “우주최강염소”를 받는다.

뱉는다의 개념 - 예제 2

```
a = "우주최강염소"
print("당신의 이름은", "우주최강염소",
```

```
>>> %Run hello.py
 이름을 입력하세요.우주최강염소
 당신의 이름은 우주최강염소 입니다.
```

뱉는다의 개념 - 예제 3

```
a = input("숫자를 입력해주세요.")
print(a, "의 두배는", int(a)*2, "입니다.")
```

```
>>> %Run hello.py
 숫자를 입력해주세요.
```

뱉는다의 개념 - 예제 3

```
a = "5"
print(a, "의 두배는", int(a)*2, "입니다.")
```

```
>>> %Run hello.py
 숫자를  입력해주세요.5
```

엔터를 치는 순간 input() 함수가 “5” 를 뱉는다.

뱉는다의 개념 - 예제 3

```
a = "5"
print("5", "의 두배는", int(a)*2, "입니다.")
```

```
>>> %Run hello.py
 숫자를  입력해주세요.5
```

엔터를 치는 순간 input() 함수가 “5” 를 뱉는다.
a는 “5”를 뱉는다.

뱉는다의 개념 - 예제 3

```
a = "5"
print("5", "의 두배는", int("5")*2, "입니다."
```

```
>>> %Run hello.py
 숫자를  입력해주세요.5
```

엔터를 치는 순간 input() 함수가 “5” 를 뱉는다.
a는 “5”를 뱉는다.

뱉는다의 개념 - 예제 3

```
a = "5"
print("5", "의 두배는", 5*2, "입니다.")
```

```
>>> %Run hello.py
 숫자를  입력해주세요.5
```

엔터를 치는 순간 input() 함수가 “5” 를 뱉는다.
a는 “5”를 뱉는다. int(“5”) 는 5 를 뱉는다.

뱉는다의 개념 - 예제 3

```
a = "5"
print("5", "의 두배는", 10, "입니다.")
```

```
>>> %Run hello.py
 숫자를  입력해주세요.5
```

엔터를 치는 순간 input() 함수가 “5” 를 뱉는다.
a는 “5”를 뱉는다. int(“5”) 는 5 를 뱉는다.
5*2 는 10 을 뱉는다.

뱉는다의 개념 - 예제 3

```
a = "5"
print("5", "의 두배는", 10, "입니다.")
```

```
>>> %Run hello.py
숫자를 입력해주세요.5
5 의 두배는 10 입니다.
```

뱉기와 출력은 다른 것!

input(“**질문**”) 함수는 **사용자의 입력**을 뱉는다.

print(A) 함수는 **A**를 출력한다.

뱉기와 출력은 다른 것!

input(“**질문**”) 함수는 **사용자의 입력**을 뱉는다.

➡ **“질문”을 출력하지 사용자의 입력은 출력하지 않음.**

print(A) 함수는 **A**를 출력한다.

뱉기와 출력은 다른 것!

input(“**질문**”) 함수는 **사용자의 입력**을 뱉는다.

➡ **“질문”을 출력하지 사용자의 입력은 출력하지 않음.**

print(A) 함수는 **A**를 출력한다.

➡ **A를 출력하지만 아무것도 뱉지 않음.**

뱉기와 출력은 다른 것!

input(“**질문**”) 함수는 **사용자의 입력**을 뱉는다.

➡ “질문”을 출력하지 사용자의 입력은 출력하지 않음.

print(A) 함수는 **A**를 출력한다.

➡ A를 출력하지만 아무것도 뱉지 않음.
a = print(“안녕하세요!”) 를 하면 a에 아무것도 들어가지 않는다.

뱉기와 출력은 다른 것!

input(“**질문**”) 함수는 **사용자의 입력**을 뱉는다.

➡ “질문”을 출력하지 사용자의 입력은 출력하지 않음.

print(A) 함수는 **A**를 출력한다.

➡ A를 출력하지만 아무것도 뱉지 않음.
a = print(“안녕하세요!”) 를 하면 a에 아무것도 들어가지 않는다.

뱉기 =/= 출력

직접 해보기

1. print(“안녕?”) 함수를 불러서 뱉는 값을 a에 넣어보자. a를 출력해 무슨 값이 들어있는지 알아보자. print 함수는 값을 뱉는가? print 함수는 값을 출력하는가?

2. input() 함수를 불러서 뱉는 값을 a에 넣어보자.a를 출력해 무슨 값이 들어있는지 알아보자. input 함수는 값을 뱉는가? input 함수는 값을 출력하는가?

직접 해보기

3. 아래와 같이 단어를 알려주는 한→영 사전을 만들어보자. 만든 후에 뱉기의 의미를 잘 생각하며 코드를 한 줄로 줄여보자.

   힌트: 딕셔내리(사전)과 input 함수를 이용해 사전을 만들자.

```
>>> %Run a.py
한글 단어를 입력해주세요.사과
영어로 apple 입니다.
```

10주차 복습

time.py

"시간"에 관련된 일을 해주는 모듈

지금이
몇시 몇분 몇초?

5초 기다리기

오늘의 날짜는?

time 모듈 불러오기

```
import time
print("time 모듈 준비 완료!")
```

여느 모듈처럼 import 를 이용해 불러와주면 된다.

time.sleep()

time.sleep(A)

A 초 만큼 기다린다.

```
>>> import time
>>> time.sleep(5)
>>> time.sleep(0.3)
```

(5초간 멈춘다.)

(0.3초간 멈춘다.)

time.sleep() 예제

```
import time
for a in range(1, 11):
    print(a, "초...")
    time.sleep(1)
```

time.sleep() 예제

```
import time
for a in range(1, 11):
    print(a, "초...")
    time.sleep(1)
```

```
>>> %Run test.py
1 초...
2 초...
```

1부터 10까지
천천히 센다.

time.time()

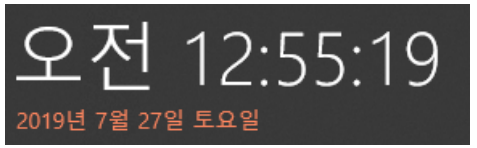
time.time()

1970년 1월 1일부터 지금까지 지난 초를 뱉는다.

```
>>> print(time.time())
1564156507.3043346
```

time.time()

```
>>> print(time.time())
1564156507.3043346
```



1970년 1월 1일부터 선생님이 이 함수를 부른 순간까지

15억 6415만 6507 초가 지났다.

time.localtime()

time.localtime()

현재 시간을 뱉는다.

```
>>> a = time.localtime()
>>> a.tm_min
52
```

time.localtime() 예제

```
import time
t = time.localtime()
year = t.tm_year
print("지금은", year, "년입니다.")
```

```
>>> %Run test.py
 지금은 2019 년입니다.
```

range - 2

range(a, b, c)

“a부터 b까지 c 간격으로 뛰어 세는 정수”
c는 쓰지 않으면 기본적으로 1이다.

```
>>> for a in range(2, 100, 2):
        print(a, "는 짝수입니다.")
```

range 예제

```
for a in range(0, 10, 2):
    print(a)
```

range 예제

```
for a in range(0, 10, 2):
    print(a)
```

```
>>> %Run test.py
0
2
4
6
8
```

range 예제

```
for a in range(5, 0, -1):
    print(a)
```

range 예제

```
for a in range(5, 0, -1):
    print(a)
```

```
>>> %Run test.py
5
4
3
2
1
```

직접 해보기

1. 현재 시간을 출력하는 프로그램을 만들자.

힌트: time.localtime() 과 그 속의 변수들을 이용하자.

```
>>> %Run test.py
현재 시간은 13 시 53 분 12 초입니다.
```

참고

tm_hour	시가 들어있음
tm_min	분이 들어있음
tm_sec	초가 들어있음
tm_wday	요일(0~6)이 들어있음

직접 해보기

2. 로켓 카운트다운을 만들어보자. **10**부터 **1**까지 **-1**의 간격으로 세다가 마지막에 **“발사!”** 를 출력하자. 단 1초 간격으로 천천히 세야한다.

힌트: for문과 range(a, b, c), time.sleep() 함수를 사용하자.

```
>>> %Run chicken.py
10 ...
9 ...
        (생략)
2 ...
1 ...
발사!
```

직접 해보기

3. **3**부터 **99**까지 3의 배수를 모두 출력해보자.

힌트: for문과 range(a, b, c) 함수를 사용하자.

```
>>> %Run hello.py
3
6
9
12
15
18
21
24
```

세계 속 파이썬

파이썬이 쓰인 게임들엔 뭐가 있을까?

Frets on Fire

“Music Video” Game

https://www.youtube.com/watch?v=c5i6SxSAY4Q

The Sims 4

Modding is done with Python to modify game mechanics

https://www.youtube.com/watch?v=tn0hCTyj2Kc

The Sims 4

World of Tanks

Scripting Language for Game Logics

https://www.youtube.com/watch?v=4Kmyj5iV98U

EVE Online

Both Server and Client made with Python
Stackless Python - Excels in Bulk Operations

https://www.youtube.com/watch?v=ZdoQzmzg2XY

세계 속 파이썬

데이터 분석 & 시각화

데이터 분석 & 시각화

미국 몬트리얼 도시의 교통 흐름을 2D 히스토그램으로 시각화

https://www.reddit.com/r/dataisbeautiful/comments/7p14gg/how_to_visualize_traffic_flow_with_dynamic_2d/

데이터 분석 & 시각화

1년간 자신의 모든 이동 경로를 동영상으로 제작

https://www.reddit.com/r/dataisbeautiful/comments/7ub3he/wrote_a_python_script_to_map_out_everywhere_i_had/

데이터 분석 & 시각화

자신이 1년간 내뿜은 CO2의 양과 이동을 시각화

https://www.reddit.com/r/dataisbeautiful/comments/cjb7up/oc_the_air_we_breathe_tracking_my_co2_emissions/

파이썬과 친해지기

뱉는다(반환)의 개념, 10주차 복습, 세계 속 파이썬

12주차

References

왕초보를 위한 Python 2.7 https://wikidocs.net/145
뱀 인형 이미지 https://bit.ly/2WlNL65
Stock Photos https://unsplash.com/