

» 학습목표

- 파이썬 프로그램의 random 모듈로 임의의 수 뽑기, while 명령으로 반복하기, 함수를 정의하고 호출하기, 함수 응용하기를 익혀 원하는 응용프로그램 만들 수 있어야 한다.

Unit1. random 모듈로 임의의 수 뽑기

01 random 모듈이란?

슬라이드 1. random 모듈이란?

- » 임의의 수(컴퓨터가 아무렇게나 생성한 무작위의 수)를 뽑는 기능을 구현하는 기능
- » `range(a, b)`와 `random.randint(a, b)`
 - `range(a, b)` : for 반복문에서 a부터 b 직전까지의 정수(즉, b-1)를 하나씩 반복할 때 사용하는 명령어
 - `random.randint(a, b)` : a부터 b까지의 임의의 정수를 만들어 내는 명령어
 - 차이점 : `range`는 b가 제외되고 `randint`는 b가 포함됨

1. random 모듈이란?

» 마음대로 걷는 거북이 1

```
» import turtle as t
import random
```

```
t.shape("turtle")
t.speed(0)
```

```
for x in range(500):
    a = random.randint(1, 360)
    t.setheading(a)
    t.forward(10)
```

'거북이' 모양의 거북이 그래픽을 사용

거북이를 500번 움직임

1~360에서 아무 수나 골라 a에 저장

거북이 방향을 a 각도로 돌림

거북이가 10만큼 앞으로 이동

실습



1. random 모듈이란?

» 마음대로 걷는 거북이 2

```
» import turtle as t  
import random
```

```
t.shape("turtle")  
t.speed(0)
```

'거북이' 모양의 거북이 그래픽을 사용

```
for x in range(500):  
    a = random.randint(1, 360)  
    t.setheading(a)  
    b = random.randint(1, 20)  
    t.forward(b)
```

거북이를 500번 움직임

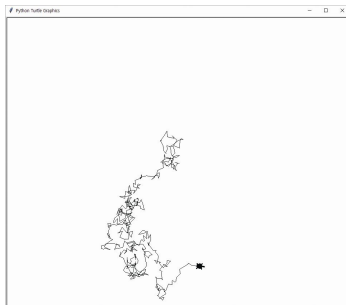
1~360에서 아무 수나 골라 a에 저장

거북이 방향을 a 각도로 돌림

1~20 사이에 있는 아무 수나 골라 b에 저장

10을 b로 고침

실습



1. random 모듈이란?

» 무작위로 덧셈 문제를 만들어서 맞추는 프로그램

» `import random`

```
a = random.randint(1, 30)    # a에 1~30 사이의 임의의 수를 저장  
b = random.randint(1, 30)    # b에 1~30 사이의 임의의 수를 저장
```

```
print(a, "+", b, "=")      # 문제를 출력  
x = input()                  # 답을 입력받아 x에 저장(문자열로 저장됨).  
c = int(x)                    # 비교를 위해 문자열을 정수로 바꿈
```

```
if a + b == c:  
    print("천재!")  
else:  
    print("바보!")
```

» 실행결과

```
21 + 2 =  
23  
천재!
```

Unit 2 .while 명령으로 반복하기

- 01** while 명령으로 반복해서 숫자를 출력하는 프로그램
- 02** 1부터 10까지 숫자의 합계를 구하는 프로그램
- 03** 숫자를 추측해서 맞히는 프로그램

슬라이드 1. while 명령으로 반복해서 숫자를 출력하는 프로그램

```
» print("[1-10]")
```

```
    x = 1
```

```
    while x <= 10:      # x가 10 이하인 동안 반복(1에서 10까지 실행)
```

```
        print(x)
```

```
        x = x + 1      # x에 1을 더해서 저장
```

» 실행결과

1

2

3

4

5

6

7

8

9

10

슬라이드 2. 1부터 10까지 숫자의 합계를 구하는 프로그램

```
» s = 0                                # 합계를 구하는 변수 s, 처음 값은 0을 입력
  x = 1
  while x <= 10:                        # x가 10보다 작거나 같은 동안, 즉 10까지 실행
      s = s + x                          # s에 x를 더함.
      print("x:", x, " sum:", s)        # 현재 x 값과 s 값을 출력
      x = x + 1                         # x에 1을 더함.
```

» 실행결과

```
x: 1 sum: 1
x: 2 sum: 3
x: 3 sum: 6
x: 4 sum: 10
x: 5 sum: 15
x: 6 sum: 21
x: 7 sum: 28
x: 8 sum: 36
x: 9 sum: 45
x: 10 sum: 55
```

2. 1부터 10까지 숫자의 합계를 구하는 프로그램

for 명령어를 사용했을 때	while 명령어를 사용했을 때
<pre>s = 0 for x in range(1, 11): s = s + x print("x:", x, "sum:", s)</pre>	<pre>s = 0 x = 1 while x <= 10: s = s + x print("x:", x, "sum:", s) x = x + 1</pre>

슬라이드 3. 숫자를 추측해서 맞추는 프로그램

```
>> import random
```

```
n = random.randint(1, 30)
```

1~30 사이에 있는 임의의 수를 뽑음

```
while True:
```

영원히 반복

```
    x = input("맞혀 보세요? ")
```

```
    g = int(x)
```

입력받은 값을 비교할 수 있도록 정수로 바꿈

```
    if g == n:
```

사용자가 추측한 값과 임의의 수가 같아야 함

```
        print("정답")
```

```
        break
```

정답을 맞히면 break로 while 반복 블록을 빠져
나감

```
    if g < n:
```

```
        print("너무 작아요.")
```

```
    if g > n:
```

```
        print("너무 커요.")
```

3. 숫자를 추측해서 맞히는 프로그램

» 실행결과

맞혀보세요? 15

너무 커요.

맞혀보세요? 10

너무 작아요.

맞혀보세요? 12

너무 작아요.

맞혀보세요? 13

너무 작아요.

맞혀보세요? 14

정답

Unit 3. 함수를 정의하고 호출하기

- 01** 함수를 정의하고 호출하는 프로그램
- 02** 인자가 있는 함수
- 03** 결과값이 있는 함수

슬라이드1. 함수를 정의하고 호출하는 프로그램

» 함수 : 자주 사용하는 프로그램의 일부분을 블록으로 분리해서 여러 번 사용할 수 있게 해 줌

- 함수를 정의한다' : 함수가 어떤 기능을 할지 파이썬에 알려 주는 것
- '함수를 호출한다' : 만들어진 함수를 실제로 사용하는 것

```
def hello():                # hello 함수를 정의
    print("Hello Python!")
```

```
hello()                     # hello 함수를 호출
hello()
hello()
```

» 실행결과

```
Hello Python!
Hello Python!
Hello Python!
```

슬라이드 2. 인자가 있는 함수

```
def hello():                # 이름을 인자로 전달받아 Hello와 함께 출력하는 함수
    print("Hello" , name)
```

```
hello2("Justin")           # Justin을 인자값으로 넣어 hello2 함수를 호출
hello2("John")              # John을 인자값으로 넣어 hello2 함수를 호출
hello2("Mike")              # Mike를 인자값으로 넣어 hello2 함수를 호출
```

» 실행결과

```
Hello Justin
Hello John
Hello Mike
```

슬라이드 3. 결과값이 있는 함수

```
» def square(a):  
    c = a * a  
    return c
```

a의 제곱(a*a)을 구하는 함수

```
def triangle(a, h):  
    c = a * h / 2  
    return c
```

밑변이 a이고 높이가 h인 삼각형의 넓이를 구하는 함수

```
s1 = 4  
s2 = square(s1  
print(s1, s2)
```

s1(4)의 제곱을 구하는 함수를 호출해 결과를 s2에 저장

```
print(triangle(3, 4)
```

밑변이 3이고 높이가 4인 삼각형의 넓이를 출력

» 실행결과

```
4 16  
6.0
```


3. 결과값이 있는 함수

» 인자는 종류가 두 가지예요?

다음 프로그램을 잠깐 볼까요?

```
def square(n):  
    return n*n
```

```
print(square(3))
```

이 프로그램은 어떤 수를 변수 n 으로 전달받아 n 의 제곱값($n*n$)을 결과로 돌려주는 함수인 `square`를 정의한 후, 이 함수에 3이라는 값을 넣어서 호출해 출력하는 프로그램입니다. 여기서 `square` 함수를 정의할 때 사용한 n 과 `square` 함수를 호출할 때 사용한 3은 둘 다 '인자'입니다.

n 과 같이 함수에서 사용되는 값을 정의하는 인자를 '형식 인자' 또는 '매개변수'라 하고, 3과 같이 함수를 호출할 때 실제로 사용되는 값을 '실 인자' 또는 그냥 '인자/인수'라고 부릅니다. 조금 복잡하죠? 엄밀하게 말하면 조금 다른 개념이지만, 처음 프로그래밍을 배우는 단계에서는 이 둘을 구분하면 오히려 혼란스러울 수 있습니다. 따라서 이 책에서는 그냥 '인자'라고 부르겠습니다.

Unit 4 . 함수 응용하기

- 01** 1부터 n 까지의 합을 구하는 함수
- 02** 1부터 n 까지 곱을 구하는 함수
- 03** 다각형을 그리는 함수

슬라이드1. 1부터 n까지의 합을 구하는 함수

```
» def sum_func(n):  
    s = 0                                # 합을 구하기 위한 변수 s(시작 값을 0으로 지정)  
    for x in range(1, n+1):              # range(1, n+1)로 1, 2, ..., n까지 반복(n+1은 제외)  
        s = s+x                          # 지금까지 계산된 s 값에 x를 더해서 다시 s에 저장  
    return s                             # 계산된 s 값을 결과값으로 돌려줌  
  
print(sum_func(10))  
print(sum_func(100))
```

» 실행결과

55

5050

슬라이드 2. 1부터 n까지 곱을 구하는 함수

» `def factorial(n):`

`fact = 1`

`for x in range(1, n+1):`

`fact = fact * x`

`return fact`

곱을 구하기 위한 변수 fact(시작 값을 1로 지정).

range(1, n+1)로 1, 2, ..., n까지 반복(n+1은 제외)

지금까지 계산된 값에 x를 곱해 fact에 다시 저장

계산된 fact 값을 돌려줌

`print(factorial(5))`

`print(factorial(5))`

» 실행결과

120

3628800

2. 1부터 n까지 곱을 구하는 함수

» 팩토리얼이 뭐예요?

함수 이름으로 사용된 factorial의 개념을 이미 알고 있는 사람도 있을 것입니다. 1부터 n까지의 양의 정수를 모두 곱한 것을 수학에서는 n 토리얼(Factorial)이라 부르며, 느낌표 기호(!)를 사용해서 n!로 표시합니다. '계승'이라고 부르기도 합니다. 예를 들어 볼까요?

- 2 팩토리얼 : $2! = 1*2 = 2$
- 5 팩토리얼 : $5! = 1*2*3*4*5 = 120$
- 10 팩토리얼 : $10! = 1*2*3*4*5*6*7*8*9*10 = 3628800$
- 단, 0!은 1이라고 약속합니다.

팩토리얼은 중고등학교 수학에서 순열이나 확률을 배울 때 자주 사용하는 계산법이니 몰랐던 사람은 이 정도로만 기억해 두세요.

슬라이드 3. 다각형을 그리는 함수

```
» import turtle as t
```

```
def polygon(n):  
    for x in range(n):  
        t.forward(50)  
        t.left(360/n)
```

n번 반복
거북이를 50만큼 앞으로 이동
거북이를 360/n만큼 왼쪽으로 회전

```
def polygon2(n, a):  
    for x in range(n):  
        t.forward(a)  
        t.left(360/n)
```

n번 반복
거북이를 a만큼 앞으로 이동
거북이를 360/n만큼 왼쪽으로 회전

```
polygon(3)  
polygon(5)
```

삼각형을 그림
오각형을 그림

```
# 그림을 그리지 않고 거북이를 100만큼 이동합니다.  
t.up()  
t.forward(100)  
t.down()
```

```
polygon2(3, 75)  
polygon2(5, 100)
```

한 변이 75인 삼각형을 그림
한 변이 100인 오각형을 그림

3. 다각형을 그리는 함수

» 실행결과

