오늘의 강의 목표

- Number에 대한 이해
- Mathematical Function들에 대한 이해
- Statistical Function들에 대한 이해
- Random Number Generator에 대한 이해

Python Number Types

- int
  - 정수
- float
  - 실수
- complex
  - 복소수

Built-in Functions for Math

- `abs(x)` : *x*의 절대값. 복소수의 경우 크기 (magnitude)

```
>>> abs(-3)
3
>>> abs(-3.14)
3.14
>>> abs(3 + 4j)
5.0
```

- `pow(x, y[, z])` : *x*의 *y*승 혹은 (*z* 존재시) 이를 *z*로 나눈 나머지

```
>>> pow(2, 4)
16
>>> pow(2, 4, 10)
6
```

Built-in Functions for Math

- min(…) : 최소값

```
>>> min(4, 2, 18, 9)
2
>>> min('pineapple', 'blueberry', 'strawberry', 'apple')
'apple'
>>> min([3, 7, 19, -1])
-1
```

- max(…) : 최대값

```
>>> max(4, 2, 18, 9)
18
```

Built-in Functions for Math

- sum(…) : 합

```
>>> a = [1, 2, 3, 4.5]
>>> sum(a)
10.5
>>> sum(a, 10)
20.5
>>> sum(range(1, 101))
5050
>>> sum(1, 2)
Traceback (most recent call last):
  File "<pyshell#51>", line 1, in <module>
    sum(1, 2)
TypeError: 'int' object is not iterable
```

Starting value

10 + sum(a)

에러 발생

Built-in Functions for Math

- round(…) : 반올림

```
>>> round(3.141592)
3
>>> round(3.141592, 3)
3.142
>>> round(3.141592, 4)
3.1416
```

소수점 이하 자리수

- divmod(…) : 몫과 나머지

```
>>> divmod(19, 3)
(6, 1)
>>> divmod(19, 0)
Traceback (most recent call last):
  File "<pyshell#61>", line 1, in <module>
    divmod(19, 0)
ZeroDivisionError: integer division or modulo by zero
```

에러 발생

More Python Built-in Functions

https://docs.python.org/3.4/library/functions.html

		Built-in Functions		
abs()	dict()	help()	min()	setattr()
all()	dir()	hex()	next()	slice()
any()	divmod()	id()	object()	sorted()
ascii()	enumerate()	input()	oct()	staticmethod()
bin()	eval()	int()	open()	str()
bool()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	
delattr()	hash()	memoryview()	set()	

Mathematical Functions

- 고급 수학 함수 사용 → import math
- 전체 목록 : https://docs.python.org/3/library/math.html

```
>>> import math
>>> math.factorial(5)
120
>>> math.sqrt(9)
3.0
>>> math.sin(0)
0.0
```

Mathematical Functions

- `ceil(x)` : Ceiling 값

```
>>> math.ceil(3.14)
4
```

- `floor(x)` : Floor 값

```
>>> math.floor(3.14)
3
```

- `factorial(x)` : Factorial 값

```
>>> math.factorial(5)
120
```

More Mathematical Functions

```
>>> help("math")
Help on built-in module math:

NAME
    math

DESCRIPTION
    This module is always available.  It provides access
to the
    mathematical functions defined by the C standard.

FUNCTIONS
    acos(...)
        acos(x)

        Return the arc cosine (measured in radians) of x.
…
```

Statistical Functions

- mean() : 평균
- median() : 중간값
- pstdev() : 표본표준편차
- pvariance() : 표본분산
- stdev() : 샘플표준편차
- variance() : 샘플분산

Full list at
https://docs.python.org/3.4/library/statistics.html

Statistical Functions

```
>>> import statistics
>>> a = [-1, 3, 99, -2, 35]
>>> statistics.mean(a)
26.8
>>> statistics.median(a)
3
>>> statistics.stdev(a)
43.15321540742938
>>> statistics.variance(a)
1862.2
```

Random Number Generator

- 난수 발생 → import random
- https://docs.python.org/3.4/library/random.html

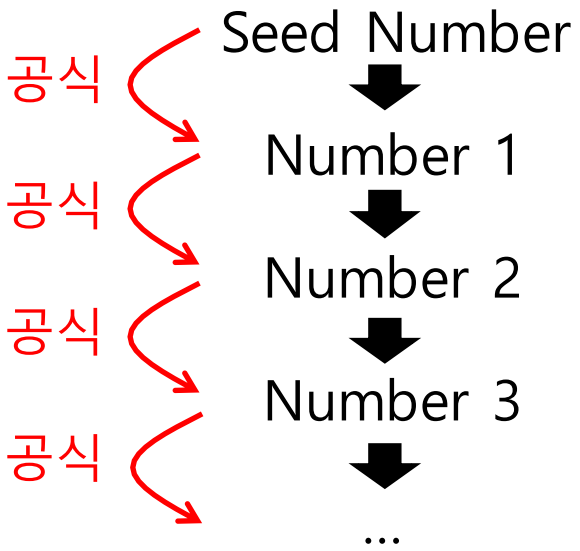
```
>>> import random
>>> for i in range(5):
      print(random.random())


0.9895525556285152
0.455022291682887
0.15583150031169168
0.794304463416605
0.4017534217902945
```

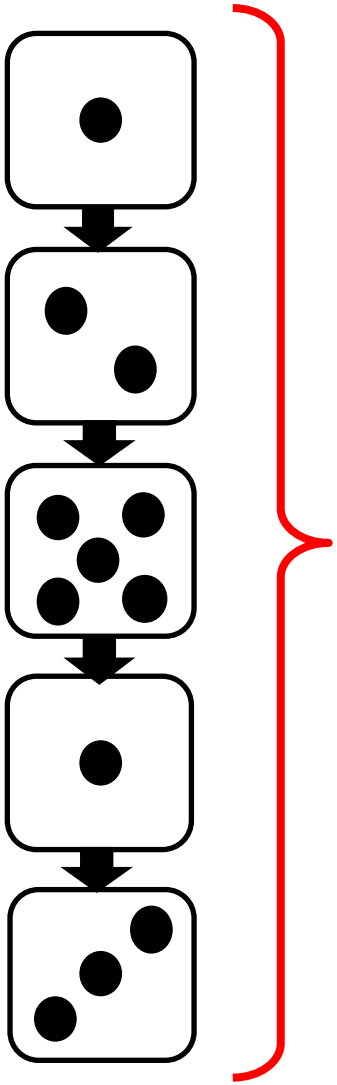
Pseudo Random Numbers within [0.0, 1.0)

Pseudo Random Number

- Pseudo Random Number Generator의 원리



Seed Number를 알면
생성될 번호 예측 가능



Random처럼 보이지만
미리 정해진 순서

Pseudo Random Number

- Seed Number가 같으면 같은 번호 생성

```
>>> random.seed(100) ← Seed Number를 100으로
>>> random.random()
0.145669255104130 3
>>> random.random()
0.4549270045140213 5
>>> random.seed(100) ← Seed Number를 100으로
>>> random.random()
0.145669255104130 3
>>> random.random()
0.4549270045140213 5
```

Pseudo Random Number

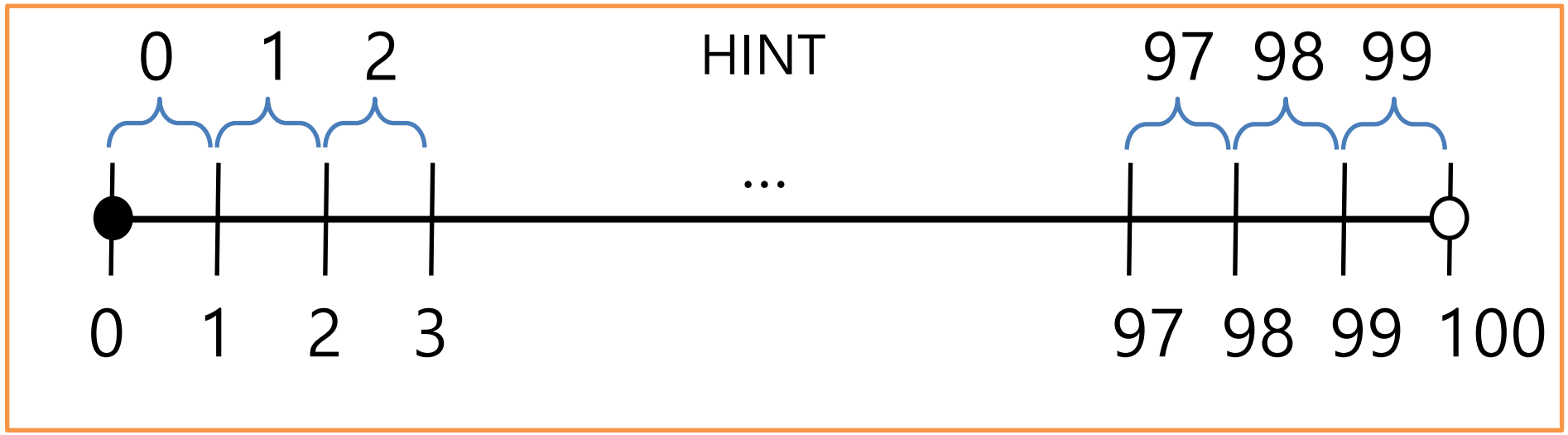
- Seed Number를 주지 않으면?
  - import 시점의 system time을 seed로 설정
  - import 시점을 알 수 있으면 예측 가능

```
>>> import random
>>> random.random()
0.770783805659022 2
```

Random Number Generator

- random.random()을 이용하여 [0, 99] 사이의 Random Integer 발생

```
>>>
```



Random Number Generator

- random.random()을 이용하여 [0, 2] 사이의 Random Integer 발생

```
>>>
```

0: 0, 3, 6, ..., 99 (34개)
1: 1, 4, 7, ..., 97 (33개)
2: 2, 5, 8, ..., 98 (33개)

Random Number Functions

- seed()
- random()
- randint()
- choice()
- shuffle()
- randrange()

Full list at
https://docs.python.org/3.4/library/random.html

Random Number Functions

- `seed(a)` : Seed를 a로 설정

```
>>> random.seed(1)
```

- `random()` : [0, 1) 사이의 실수 발생

```
>>> random.random()
0.770783805659022 2
```

- `randint(a, b)` : [*a*, *b*] 사이의 정수 발생

```
>>> random.ranint(3, 33)
7
```

Random Number Functions

- choice(*seq*) : *seq* 중에 랜덤하게 선택

```
>>> random.choice(['red', 'green', 'blue'])
'green'
```

- shuffle(*seq*) : *seq* 섞음

```
>>> a = ['red', 'green', 'blue']
>>> random.shuffle(a)
>>> a
['blue', 'green', 'red']
```

- randrange(*a*[, *b*, *c*]) : choice(range(*a*, *b*, *c*)) 와 동일

```
>>> random.randrange(1, 40, 7)
15
```

Advanced Topics

- Floating-point
  - https://docs.python.org/3.4/tutorial/floatingpoint.html

```
>>> 3 % 1.2
0.6000000000000001
```

Questions