

Python

Python Programs

- a program is a sequence of definitions and commands
프로그램은 정의와 명령의 연속임
 - definitions evaluated 정의에 따라 평가됨
 - commands executed by Python interpreter in a shell 명령은 해석기가 실행함
- commands (statements) instruct interpreter to do something
명령에 따라 해석기가 동작함
- can be typed directly in a shell or stored in a file that is read into the shell and evaluated
셸에서 직접 쓸 수도 있고, 파일에 저장해서 읽히도록 할 수도 있음

Objects

- Python program manipulates data objects 객체를 조작함
- Objects have a type that defines the kinds of things program can do to them
객체에 프로그램에 정의된 여러 동작을 적용할 수 있음
- objects are 객체는 수와 수가 아닌 것을 포함함
 - scalar (cannot be subdivided)
 - non-scalar

Scalar objects

- int – represent integers, ex. 1, 2, 3, 4, etc. 정수
- float – represent real numbers, ex. 3.14, 48.12 실수
- bool – represent Boolean values True and False 논리 값
- can use type() to see the type of an object type이라는 명령어로 형을 알 수 있음
 - >>> type(5)
int
 - >>> type(3.14)
float

Type conversion (cast)

- can convert object of one type to another 자신의 형이 아닌 다른 형으로 변환 가능함
- example: 예
 - `float(3)` converts integer 3 to `float(3.0)` 정수를 실수로 바꿈
 - `int(3.9)` truncates float 3.9 to integer 3 실수를 정수로 바꿈

Printing to console

- to show output from code to a user, use print command print 명령으로 출력할 수 있음

In [30]: 3+8

Out [30]: 11

“out” tells you it’s an interaction within the shell only

In [31]: print(3+8)

11

No “out” means it is actually shown to a user when you run a file

Expressions

- combines objects and operators to form expressions 연산자와 객체를 조합하여 표현식을 만들 수 있음
- an expression has a value, which has a type 표현식은 형을 갖는 값
- syntax for a simple expression 간단한 표현식의 문법
 - <object> <operator> <object>

Operators for int and float types

- $i + j \rightarrow$ sum, int \rightarrow int, float \rightarrow float
- $i - j \rightarrow$ difference , int \rightarrow int, float \rightarrow float
- $i * j \rightarrow$ product , int \rightarrow int, float \rightarrow float
- $i / j \rightarrow$ division, result is always float

- $i \% j \rightarrow$ the modular operator, it gives remainder when I is divided by j
나머지 연산자,
- $i ** j \rightarrow$ I to the power of j 승을 계산

Binding variables and values

- equal sign is an assignment of a value to a variable name 등호를 써서 변수 이름에 값을 할당함

`pi = 3.141592`

`pi_approx = 22/7`

- value stored in computer memory 값은 컴퓨터 메모리에 저장됨
- an assignment binds name to value 할당으로 이름과 값을 연결
- retrieve value associated with name or variable by invoking the name, by typing pi 이름을 부르면 연결된 값을 읽어옴

Abstracting expressions

- Why give names to values of expressions? 변수를 사용하여 수식을 표현하는 이유
 - to reuse names instead of values 값이 아니라 이름으로 재사용 가능하도록
 - easier to change the code later 코드의 변경이 쉬움

```
pi = 3.141592
```

```
radius = 2.2
```

```
area = pi*(radius**2)
```