

## 2. Python 기초

### 2.1. 설치

1. 구글에서 "anaconda download"로 검색하거나 주소창에 <https://www.anaconda.com/download/> 을 입력
2. 운영체제에 맞는(예: 윈도우 64bit) Python 3.X 버전을 다운 받는다
3. 다운받은 Installer를 실행하여 설치한다. 윈도우에서는 반드시 다음과 같이 옵션을 선택한다.
  - **Install for** 옵션에서는 **Just Me** 선택
  - **Advanced Options**에서는 "Add Anaconda ...", "Register Anaconda ..." 옵션을 모두 체크한다.
  - **VS Code** 설치는 **Skip**을 선택

### 2.2. Jupyter Notebook 실행

jupyter notebook은 파이썬 코드를 편집하고 실행하는 도구이다. 다음과 같은 방법으로 실행한다.

윈도의 경우 다음과 같이 한다.

1. 탐색기로 원하는 작업 폴더를 연다.
2. 탐색기 주소창에 `jupyter notebook` 을 입력한다.

맥의 경우에는 다음과 같이 한다.

1. Terminal을 연다.
2. `cd` 명령으로 원하는 작업 폴더로 이동한다.
3. `jupyter notebook` 을 입력한다.

### 2.3. 값

파이썬에서는 수, 문자 등의 값(value)을 다룰 수 있다. 값이란 프로그램에서 다루는 대상을 말한다.

값을 입력하면 그 결과를 보여준다.

```
3.14
```

```
3.14
```

문자열은 앞뒤로 따옴표를 붙여 표시한다.

```
'hello'
```

```
'hello'
```

입력된 값을 계산하거나 다양한 방식으로 다룰 수 있다.

```
1 + 3
```

```
4
```

## 2.4. 변수

변수는 값에 이름을 붙이는 것이다. 파이썬에서는 알파벳 대소문자, 숫자, 밑줄( `_` )로 변수명을 지을 수 있다. 단, 변수 이름 첫 글자에는 숫자를 쓸 수 없으며, 알파벳은 대소문자를 구분한다. 한글 등도 쓸 수 있다.

```
a = '안녕 세상'
```

```
a
```

```
'안녕 세상'
```

## 2.5. 리스트

파이썬에서는 여러 개의 값을 다루기 위한 다양한 자료 구조를 지원한다. 가장 많이 쓰이는 것은 리스트(list)이다. 리스트는 여러 개의 값을 순서대로 저장할 때 사용한다.

```
text = ['안녕 세상', '사과 맛있다']
```

리스트의 값들은 인덱스를 통해 접근할 수 있다. 인덱스는 0부터 시작한다.

```
text[0]
```

```
'안녕 세상'
```

```
text[1]
```

```
'사과 맛있다'
```

## 2.6. 함수

함수는 값을 처리하는 일종의 명령이다. 함수 이름 뒤에는 괄호를 붙여 값을 넘겨준다. 함수는 넘겨 받은 값을 처리하고 그 결과를 다시 돌려준다.

```
len(a) # 변수 a가 가리키는 문자열의 길이를 돌려준다
```

```
5
```

### 2.6.1. 메소드

비슷한 일을 하지만 값의 종류에 따라 다른 방식으로 작동해야하는 함수들이 있다. 이런 함수들은 메소드라는 형식으로 만들어지는데 값.메소드 형식으로 부른다.

```
a.split() # 공백을 기준으로 변수 a가 가리키는 문자열을 자른다.
```

```
['안녕', '세상']
```

## 2.7. 모듈

비슷한 기능들을 모아둔 것을 모듈(module)이라고 한다. 파이썬에서는 여러 가지 모듈들을 불러와서 기능을 추가할 수 있다.

아래는 수학 계산과 관련된 `math` 모듈을 불러오는 코드다.

```
import math
```

모듈 안의 함수를 부를 때는 `모듈명.함수` 의 형태로 부른다. 다음은 `math` 모듈의 `sqrt` 함수로 4의 제곱근을 구하는 코드이다.

```
math.sqrt(4)
```

```
2.0
```

자주 사용하는 모듈의 경우 이름이 길면 타이핑 하기 어려우므로 임포트할 때 `as` 를 사용해 별명을 붙이기도 한다.

```
import math as m # math 모듈을 m이라는 별명으로 불러온다  
m.sqrt(9)
```

```
3.0
```

자주 사용하는 함수는 `from .. import ..` 를 사용해 모듈에서 바로 임포트할 수도 있다. 이 경우에는 모듈명을 붙여주지 않아도 된다.

```
from math import cos # math 모듈에서 cos 함수를 불러온다  
cos(0) # 코사인 0을 구한다
```

```
1.0
```

특정 모듈의 모든 함수를 불러올 때는 `from .. import *` 를 사용한다.

```
from math import * # math 모듈에서 모든 함수를 불러온다  
sin(0) # 사인 0을 구한다
```

```
0.0
```

`from .. import ..` 를 사용하면 함수 이름이 겹칠 때는 나중에 임포트한 함수가 먼저 임포트한 함수를 덮어 쓸 수 있으므로 주의가 필요하다.