

# 파이썬 한 입에 쏙!

담당교수 : 노 경 속  
puresue113@google.com

参考

## 1차시

## 파이썬 기초 문법 익히기

## 파이썬 기초 문법

- 변수와 자료형
- 입출력
- 조건문 (if / elif / else)
- 반복문 (for / while / break)
- 구구단, 짝수·홀수 판별
- 간단한 계산기 프로그램 만들기

## 리스트와 딕셔너리

- 리스트와 딕셔너리 개념 및 기본 문법 학습
- 반복문과 조건문을 활용한 자료 처리 실습
- 간단한 한국어-중국어 사전 만들기
- 자판기 프로그램(메뉴 선택, 금액 처리) 구현
- 학생 성적 관리 프로그램(평균 최고·최저 점수 계산)

## 데이터분석 기초

- matplotlib을 활용한 그래프 시각화 (선 그래프, 막대 그래프, 산점도, 히스토그램)
- CSV 파일 불러오기와 저장하기
- 그래프를 이용한 데이터 분석 실습
- 간단한 웹 데이터 가져오기(예: 날씨, 주식 데이터)

## 앱 제작

- Streamlit 기본 구조와 사용자 인터페이스 이해
- 입력창, 버튼, 슬라이더 등 위젯 활용
- 데이터 시각화와 연계한 대화형 앱 만들기
- 미니 프로젝트 제작 (예: 성적 관리 대시보드, 데이터 분석 앱 등)

## 参考

## 四次课课程概要

## 1. 第1次课: 学习 Python 基础语法

Python 基础语法

变量与输入/输出 (input / print / f-string)

条件语句 (if / elif / else)

循环语句 (for / while / break)

实例: 乘法表、奇偶数判断、简单计算器

## 2. 第2次课: 列表与字典

创建字典示例 (简易韩中词典)

自动售货机示例 (结合字典与条件语句)

学生成绩管理程序

## 3. 第3次课: 数据可视化与网络爬虫

使用 matplotlib 进行图表可视化

柱状图、折线图、饼图、散点图进行数据分析

获取网络数据

## 4. 第4次课: 利用 Streamlit 制作应用

Streamlit 基本结构与界面

用户输入与按钮交互

小型项目制作

## 개발 환경

3

- 파이썬 공식 홈페이지에서 python 또는 아나콘다 설치

- 파이썬

- <https://www.python.org/downloads/>

- 아나콘다

- 파이썬에 데이터 분석 및 처리에 필요한 라이브러리들이 포함되어 있어 편리하지만 용량이 큼

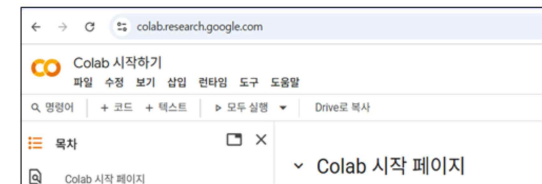
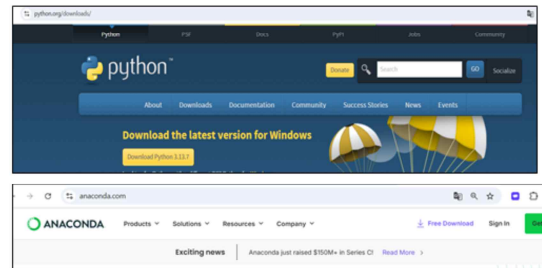
- <https://www.anaconda.com/>

- 구글 코랩

- 구글에서 제공하는 환경으로, 웹 브라우저에서 프로그램을 작성하고 실행가능함

- 다양한 라이브러리가 기본 설치되어 있으며, 작성한 코드가 구글 드라이브에 저장됨

- <https://colab.research.google.com>



## 参考

- 从 Python 官方网站下载安装 Python 或 Anaconda
- Python: <https://www.python.org/downloads/>
- Anaconda: 包含数据分析与处理所需的库, 使用方便, 但文件体积较大 <https://www.anaconda.com/>
- Google Colab: 谷歌提供的编程环境, 可以在网页浏览器中编写和运行程序。已预装多种库, 编写的代码会保存到谷歌云端硬盘 <https://colab.research.google.com>

## 학습 내용

4

- 파이썬 실행환경(IDE/노트북/터미널) 세팅과 스크립트 실행법 이해
- 자료형(int/float/str/bool)과 변수, 연산자, 입력/출력 사용
- 조건문(if/elif/else), 반복문(for/while)로 기본 문제 해결

## 参考

### 学习内容

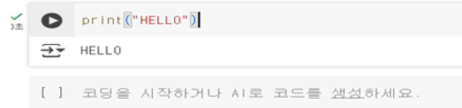
- 理解 Python 运行环境 (IDE/Notebook/Terminal) 的设置与脚本运行方法
- 使用数据类型 (int/float/str/bool)、变量、运算符、输入/输出
- 使用条件语句 (if/elif/else)、循环语句 (for/while) 解决基本问题

## 기본 코드 작성

5

### □ 코드 작성하고 실행하기

- 셀 단위로 실행을 하며 한 셀에 입력을 하거나, 여러셀에 나누어 입력 가능
- 코드 작성 후 enter : 셀에 다른 코드 추가 작성
- ctrl+enter : 실행
- shift+enter : 실행 후 셀 추가



### □ 출력 : print() 함수

- # 기호는 주석(comment)의 의미로 코드 실행 및 결과에 영향을 주지 않음

#### CODE

```
print("Hello")
print("World")
print(7+3)
print('7+3')
print("Hello "*3)
print("10"+3)
print(Hi*3)
```

```
[3] #주석
print("Hello")
print("World")
```

```
Hello
World
```

결과

```
[4] print(7+3)
print('7+3')
```

```
10
7+3
```

결과

```
print("Hello "*3)
print("10"+3)
print(Hi*3)
```

## 参考

### 基本代码编写

- 编写并运行代码
  - 按单元格执行, 可以在一个单元格中输入, 也可以分开多个单元格输入
  - 编写代码后按 Enter: 在同一单元格中继续输入代码
  - Ctrl+Enter: 执行
  - Shift+Enter: 执行并新增单元格
- 输出: print() 函数
  - # 符号表示注释 (comment), 不会影响代码执行和结果

## 변수

6

### □ 변수

- 프로그램이 실행되는 동안 값을 저장하는 공간

- `age=20`
  - `print(age)` : 변수의 내용 출력 `print(변수이름)`

- 변수이름 규칙 : 첫번째 글자는 문자 또는 밑줄(`_`), 나머지 글자는 문자, 숫자, 밑줄(`_`)가능

### □ 대입(할당)

- `=` 연산자를 사용하여 값을 변수에 저장 (자료형을 지정하지 않아도 됨)

- `x = 10`      # 정수형(int)
  - `y = 3.14`    # 실수형(float)
  - `name = "Kim"` # 문자열(str)

## 参考

- 变量：在程序运行过程中用来存储数值的空间
  - `age=20`
  - `print(age)` : 输出变量的内容 `print(变量名)`
- 变量命名规则：第一个字符必须是字母或下划线(), 其余字符可以是字母、数字或下划线()
- 赋值：使用 `=` 运算符将数值存入变量 (无需指定数据类型)
  - `x = 10` # 整型(int)
  - `y = 3.14` # 浮点型(float)
  - `name = "Kim"` # 字符串(str)

## 입력

7

- input()함수
  - ▣ 변수=input(안내문)
- int() 함수
  - ▣ ()안의 값을 정수로 변환
  - ▣ int(변수 또는 숫자로 변환할 값)
- float()함수
  - ▣ ()안의 값을 실수로 변환

## CODE

```
a=input("number:")
b=input("숫자:")
print(a+b)
print(a-b)
TypeError
```

```
a=input("number:")
b=input("숫자:")
print(a+b)
print(a-b)

number:20
숫자:10
2010

TypeError
/tmp/ipython-input-1802727428_0.py
2 b=input("숫자:")
3 print(a+b)
```



```
a=int(input("number:"))
b=int(input("숫자:"))
print(a+b)
print(a-b)

number:20
숫자:10
30
10
```

## 参考

## 输入

- input() 函数: 变量 = input(提示语)
- int() 函数: 将括号内的值转换为整数 int(变量或需要转换的数值)
- float() 函数: 将括号内的值转换为浮点数

## 출력

8

## □ print(값, 값,... 값)

## CODE

```
name = 'kim'
score = 90
age = 20
print(name, age, score)
```

```
▶ name = 'kim'
score = 90
age = 20
print(name, score, age)
print("이름은", name, "나이는", age, "코딩점수는", score, "점 입니다.")
print(f'이름은{name}이고, 나이는 {age}, 코딩점수는{score}점 입니다.')
```

```
↻ kim 90 20
이름은 kim 나이는 20 코딩점수는 90 점 입니다.
이름은kim이고, 나이는 20, 코딩점수는90점 입니다.
```

## □ f-string

- 문자열 앞에 f 또는 F를 붙이고 변수에 중괄호({ })로 감싸서 표현

## CODE

```
name = 'kim'
score = 90
age = 20
print(f'이름은 {name}이고 나이는 {age},
코딩점수는 {score}점 입니다.')
```

## 参考

## 输出

- print(值, 值, ... 值)
- f-string: 在字符串前加 f 或 F, 变量放在大括号 { } 中



## 연산자

9

- 산술 연산자
  - ▣ +, -, \*, /, //(몫), %(나머지), \*\*(거듭제곱)
- 비교 연산자
  - ▣ ==, !=, >, <, >=, <=
- 논리 연산자
  - ▣ and, or, not

## CODE 2

```
# 조건: 12세 이상 and 키 140cm 이상이면 놀이기구 탑승 가능
age = int(input("나이: "))
height = int(input("키(cm): "))
can_ride = (age >= 12) and (height >= 140)
print("탑승 가능 여부:", can_ride)
```

## CODE 1

```
print(10+3)
print(10-3)
print(10*3)
print(10/3)
print(10//3)
print(10**3)

print(10==3)
print(10!=3)
print(10>=3)
print(10<3)

print(10>3 and 10<4)
print(10>3 or 10<4)
```

## 参考

- 运算符
- 算术运算符: +, -, \*, /, //(商), %(余数), \*\*(幂运算)
- 比较运算符: ==, !=, >, <, >=, <=
- 逻辑运算符: and, or, not
- 条件: 12岁以上 并且 身高在 140cm 以上时, 可以乘坐游乐设施

## 입출력 활용하기

10

## 1. 원의 반지름 입력 후 원 면적 출력하기

## CODE 1

```
r = float(input("원의 반지름을 입력하세요: "))
area = 3.14 * (r ** 2)
# 결과 출력 (소수 둘째 자리까지 표시)
print(f"반지름 {r}의 원 넓이는 {area:.2f} 입니다.")
```

## 2. 시간을 초 단위로 입력 받아서 시간, 분, 초로 표시하기

## CODE 2

```
time = int(input("초 단위 시간:"))
hour = time//3600
min = (time%3600)//60
second = time%60
print(f'{time}초는 {hour}시간 {min}분 {second}초')
```

## 参考

## 输入输出应用

1. 输入圆的半径，输出圆的面积
2. 这个程序的功能是：把用户输入的 **秒数** 转换成 **时、分、秒**。
  - // 表示取整除（得到小时数）
  - % 表示取余数（剩下的部分再换算成分和秒）

## 선택구조(조건문)

11

- if : 조건식이 참이면 문장 실행

if 조건식:  
문장

\*주의 : 조건식뒤에 콜론(:)과 문장앞은 들여쓰기

- if~else : 조건식이 참이면 문장1, 거짓이면 문장2 실행

if 조건식:  
문장1  
else:  
문장2

- if~elif~else : 조건식이 여러 개인 경우 조건식이 참인 문장을 실행하고 나머지는 실행하지 않음

if 조건식1:  
문장1  
elif 조건식2:  
문장2  
else  
문장3

### CODE1

```
score = int(input("점수 입력"))
if score >= 90:
    grade = "A"
print(grade)
#90점 이상일때만 grade에 값이 부여됨
```

### CODE3

```
score = int(input("점수 입력"))
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
elif score >= 70:
    grade = "C"
else:
    grade = "D"
print("학점:", grade)
```

### CODE2

```
score = int(input("점수 입력"))
if score >= 60:
    grade = " 합격 "
else:
    grade = " 불합격 "
```

## 参考

### 选择结构（条件语句）

- if: 若条件为真，则执行语句
- if~else: 若条件为真，则执行语句1，否则执行语句2
- if~elif~else: 当有多个条件时，执行条件为真的语句，其余的语句不执行。

## if 활용

12

1. 정수를 입력받아서 짝수인지 홀수인지 판별하기

&lt;출력&gt;

숫자 : 7

홀수

2. 놀이공원 입장료 5000원이고 나이에 따라 할인을 적용

나이	입장료
8세 미만	기본 입장료의 30%인하
8세 이상 60세 미만	기본 입장료
60세 이상	기본 입장료의 20%인하

&lt;출력&gt;

나이: 00

입장료: 0000 원

## 参考

## if 的应用

1. 输入整数，判断是偶数还是奇数
2. 游乐园门票 5000韩元，根据年龄实行折扣

年龄	入场费
8岁以下	基本入场费的 30% 折扣
8岁以上 ~ 60岁未滿	基本入场费
60岁及以上	基本入场费的 20% 折扣

## 반복구조(for)

13

- for : 정해진 횟수만큼 반복해서 문장을 실행
  - for 임시변수 in 값:
    - 문장
  - range(시작값, 종료값, 증가값) 함수 : 시작값에서 종료값 이전까지 정수를 생성함. '증가값'은 생략하면 1이며, 증가되는 단위를 넣을 수 있음
    - range(5) : 0,1,2,3,4 #시작은 0부터, 증가값 기본값 1, 종료값 5 이전까지 생성
    - range(1,5):1,2,3,4
    - range(1,5,2):1,3
  - for a in **range(5)**:
    - print("hello")
      - range(5)에 의해 0~4까지 생성되며 a가 그 값을 하나씩 갖게됨
      - a가 0이 되고 hello 출력
      - a가 1이 되고 hello 출력
      - a가 2가 되고 hello 출력
      - a가 3이 되고 hello 출력
      - a가 4가 되고 hello 출력 **총 5회** 출력

## 参考

## 循环结构 (for)

- for : 按指定次数重复执行语句
- range(起始值, 结束值, 增加值)
  - 从起始值开始生成整数, 直到结束值之前。
  - 如果省略“步长”, 默认值为 1, 也可以设置步长来控制增加的单位
- range(5) → 0,1,2,3,4
- range(1,5) → 1,2,3,4
- range(1,5,2) → 1,3

## for 활용

14

- for i in range(10):  
    print(i, end=" ")
- for a in range(1,11,2):  
    print(a\*10)
- for k in range(10,0,-1):  
    print(k)
- sum = 0  
    for i in range(1,11):  
        sum=sum+i  
        print(sum)  
    print("총합:", sum)

参考

## 반복구조(while)

15

□ while문 : 조건식이 참인 동안 문장 반복 실행

□ while 조건식:  
반복할 문장

CODE

```
a=1
while a<=3:
    print(a)
    a = a+1
```

1  
2  
3

CODE

```
sum = 0
i = 1
while(i<11):
    sum=sum+i
    print(sum, end=" ")
    i=i+1
print("총합:", sum)
```

1 3 6 10 15 21 28 36 45 55 총합: 55

매일 100원씩 올려 5만원 모으기

```
sum = 0
money = 100
day = 0
while sum<50000:
    sum = sum+money
    day +=1 #day=day+1
    money +=100 #money=money+100
print(day)
print(sum)
print(money)
```

## 参考

- while 语句：当条件为真时，重复执行语句
- 计算从 1 到 10 的所有整数的和。
- 每天存的钱增加 100 韩元，直到存款达到 50,000 韩元为止，计算需要多少天

## 반복 구조 종료하기(break)

16

- while True :
  - if 조건 :
  - break
- 숫자게임
  - 랜덤한 숫자를 생성하기 위한 random 라이브러리의 random.randint() 사용
  - import random
  - random.함수이름
  - random.randint(시작 값, 끝 값)

## 숫자게임

```
import random
answer = random.randint(1, 100)
print("숫자 맞추기 게임 시작! (1~100 사이)")

while True:
    guess = int(input("숫자를 입력하세요: "))
    if guess == answer:
        print("정답입니다!")
        break
    elif guess < answer :
        print("UP")
    else:
        print("DOWN")
```

## 参考

## 结束循环 (break)

- while True:
  - if 条件:
  - break
- 猜数字游戏 (用户猜测随机数的游戏)

使用 **random** 库中的 **random.randint()** 来生成随机数。

调用 **random.randint(起始值, 结束值)** 可以生成在该范围内的随机整数。

- 1) 电脑在 1~100 之间随机生成一个整数
- 2) 用户输入数字来猜测
- 3) 如果比答案小 → 输出 "UP" (再大一点)
- 4) 如果比答案大 → 输出 "DOWN" (再小一点)
- 5) 如果猜中 → 输出 "答对了!" 并结束游戏



## 반복문 활용 - 구구단 출력

17

- for문과 while문을 이용하여 구구단 만들기

## CODE

```
dan = int(input("단:"))
for i in range(1,10):
    print(f'{dan}*{i}={dan*i}')

k=1
while(k<=9):
    print(f'{dan}*{k}={dan*k}')
    k=k+1
```

## 参考

循环应用 - 乘法表输出

- 使用 for 和 while 语句制作乘法口诀表

## 반복문 활용 - 구구단 게임

18

- 랜덤 값 만들기
  - ▣ random.randint(2,9)
- 정답과 입력값 비교 후 채점
  - ▣ if
- 구구단 퀴즈 반복 횟수
  - ▣ for 또는 while

## CODE

```
print("=== 구구단 퀴즈 게임 ===")
score = 0
rounds = 5 # 문제 개수
for i in range(1, rounds+1):
    a = random.randint(2, 9)
    b = random.randint(1, 9)
    answer = a * b
    print(f"\n문제 {i}: {a} x {b} = ?")
    user = int(input("정답 입력: "))
    if user == answer:
        print("정답입니다!")
        score += 1
    else:
        print("틀렸습니다. 정답은", answer)
print("\n게임 종료!")
print("맞힌 개수:", score, "/", rounds)
```

```
=== 구구단 퀴즈 게임 ===
문제 1: 8 x 4 = ?
정답 입력: 30
틀렸습니다. 정답은 32

문제 2: 6 x 2 = ?
정답 입력: 12
정답입니다!

문제 3: 7 x 8 = ?
정답 입력: 50
틀렸습니다. 정답은 56

문제 4: 6 x 3 = ?
정답 입력: 18
정답입니다!

문제 5: 5 x 1 = ?
정답 입력: 5
정답입니다!

게임 종료!
맞힌 개수: 3 / 5
```

## 参考

- 循环应用 – 乘法表游戏
  - random.randint(2,9)
- 使用 if 判断答案
- 设定题目次数 (for 或 while)
- 乘法表测验游戏:
  - 1) 程序会随机生成两个数字 (2~9 与 1~9) , 然后组成一道乘法题。
  - 2) 用户需要输入答案。
  - 3) 如果答案正确 → 输出“答对了! ”, 并加 1 分。
  - 4) 如果答案错误 → 输出“答错了, 正确答案是 ...”。
  - 5) 一共出 5 道题, 最后输出用户答对的题目数量。