

파이썬프로그래밍 2분반 ## <5주차 과제> ---

정보보안공학과 ### 202121556 ### 객지현

2023-04-09

In [1]: # 1. 문자열 내장함수 find, rfind 함수를 이해를 위해 다음 코드를 실행하여 결과를 확인해보

```
str1 = "나보안은 2021년도 1학기에 파이썬 프로그래밍 과목을 수강 중이다." # 문자열변수에
print(str1.find("2021")) # 문자열 안에서 왼쪽부터 2021을 찾아 처음 등장하는 위치를 출력
print(str1.find("1")) # 문자열 안에서 왼쪽부터 1을 찾아 처음 등장하는 위치를 출력
print(str1.rfind("1")) # 문자열 안에서 오른쪽부터 1을 찾아 처음 등장하는 위치를 출력
```

```
5
8
12
```

In [2]: # 2. in 연산자의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.

```
str1 = "나보안은 2021년도 1학기에 파이썬 프로그래밍 과목을 수강 신청했다." # 문자열변수
print("상명" in str1) # 문자열 변수안에 "상명"이 없어서 False를 출력
print("나보안" in str1) # 문자열 변수안에 "나보안"이 있어서 True를 출력
```

```
False
True
```

In [3]: # 3. 문자열 내장함수 split 함수의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.

```
str1, str2, str3 = "Hello Python Programming".split() # "Hello Python Programming"을 공
print(str1, str2, str3) # str1, str2, str3를 출력
```

```
Hello Python Programming
```

In [1]: # 4. 문자열 내장함수 lstrip, rstrip, strip 함수의 이해를 위해 다음 코드를 실행하여 결과

```
str1 = " * 5 + "strip Test" + " * 5 # 문자열 변수에 공백*5 + "strip Test"+ 공백*5 를
print("|"+str1+"|") # str1을 출력
print("|"+str1.lstrip()+"|") # 문자열 왼쪽의 공백을 제거하고 출력
print("|"+str1.rstrip()+"|") # 문자열 오른쪽의 공백을 제거하고 출력
print("|"+str1.strip()+"|") # 문자열 양쪽의 공백을 제거하고 출력
```

```
| ... strip Test ... |
|strip Test ... |
| ... strip Test|
|strip Test|
```

In [4]: # 5. format 함수의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.

```
a = 10.1 # a변수에 10.1을 대입
b = 123.123 # b변수에 123.123을 대입
print("a = %d, b = %f" %(a,b)) # a의 값은 정수로, b의 값은 실수로 출력
```

```
print("a = {}, b = {}".format(a, b)) # a의 값, b의 값 출력
print("a = {:d}, b = {:f}".format(int(a), b)) # a의 값은 정수로, b의 값은 실수로 출력
```

```
a = 10, b = 123.123000
a = 10.1, b = 123.123
a = 10, b = 123.123000
```

In [6]: # 6. 문자열 내장함수 upper, lower 함수의 이해를 위해 다음 코드를 실행하여 결과를 확인해

```
str1, str2, str3 = "Hello Python Programming".split() # "Hello Python Programming"을 공
print("{} {} {}".format(str1.upper(), str2.lower(), str3.upper())) # str1은 대문자로,
HELLO python PROGRAMMING
```

In [18]: # 7. 문자열 내장함수 isalnum, isalpha, isdigit, islower 함수의 이해를 위해 다음 코드를

```
print("a123".isdigit()) # "a123"에 알파벳 또는 숫자로만 구성되어 있어서 True를 출력
print("010-1234-1234".isdigit()) # "010-1234-1234"에 알파벳은 없고 숫자로만 구성되어 있
print("a123".isalpha()) # "a123"에 알파벳과 숫자로 구성되어 있어서 False를 출력
print("123".isdigit()) # "123"에 숫자로만 구성되어 있어서 True를 출력
print(str(chr(65)).islower()) # 문자열(65->A)이 소문자가 아닌 대문자이어서 False를 출력
```

```
True
False
False
True
False
```

In [19]: # 8. 문자열 내장함수 replace의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.

```
str1 = "010-1234-1234" # 문자열 변수에 "010-1234-1234"를 대입
print(str1.replace("-", "+")) # str1 문자열안에 "-"를 "+"로 변경하여 출력
print(str1.replace("-", " ").split()) # str1 문자열안에 "-"를 공백으로 변경하고 공백을
print("%s" % (str1.split("-"))) # "-"를 기준으로 문자를 나눠서 출력
print("{} {}".format(str1.split("-"))) # "-"를 기준으로 문자열을 나눠서 출력
```

```
010+1234+1234
['010', '1234', '1234']
['010', '1234', '1234']
['010', '1234', '1234']
```

In [17]: # 11. format() 함수를 활용하여 출력 예시와 같이 나오도록 프로그램의 빈칸을 채워 결과를

```
ori = "{}".format(10000) # ori 변수에 10000을 대입
remain = "{}".format(6000.0) # remain 변수에 6000.0을 대입

# '원래 돈 - 남은 돈' 공식을 이용하여 format()함수로 작성
americano = "{}".format(10000-6000.0) # americano에 (10000-6000.0) 값을 대입

print("나보안은 지갑 속 "+ori+"원 중 아.아를 구매하고 "+remain+"원이 남았다.") # ori와
print("따라서 나보안이 구매한 아.아는 "+americano+"원이다.") # americano의 값을 출력
```

```
나보안은 지갑 속 10000원 중 아.아를 구매하고 6000.0원이 남았다.
따라서 나보안이 구매한 아.아는 4000.0원이다.
```