

`split` 메소드는 파라미터로 넘겨주는 값을 기준으로 문자열을 분리시키고, 분리된 값들이 정리되어 있는 리스트를 리턴해줍니다. 이렇게 말해서는 이해가 잘 안 가니까 예제를 봅시다.

```
print("1. 2. 3. 4. 5. 6".split("."))
```

이렇게 하면 "1. 2. 3. 4. 5. 6" 을 "." 기준으로 분리를 시켜서 리스트를 생성합니다. 따라서 프로그램을 실행하면 아래와 같은 값이 출력됩니다.

```
['1', ' 2', ' 3', ' 4', ' 5', ' 6']
```

그런데 인덱스 1의 값부터는 숫자 앞에 띄어쓰기가 하나씩 있습니다. 기존의 문자열을 보면 두 숫자 사이에는 점과 띄어쓰기(" . ")가 모두 있기 때문이죠. 문제를 고치기 위해서는 파라미터를 "." 대신 ". "를 넘겨주면 됩니다.

```
print("1. 2. 3. 4. 5. 6".split(". "))
```

```
['1', '2', '3', '4', '5', '6']
```

파라미터가 없는 경우

사실 `split` 메소드의 파라미터는 옵션일 파라미터입니다. 아무 값도 넘겨주지 않았을 경우 기본값으로 ""가 넘어가게 됩니다. 그러면 공백(화이트 스페이스)을 기준으로 문자열을 나누게 되는거죠.

```
print("1 2 3 4 5 6".split())
```

```
['1', '2', '3', '4', '5', '6']
```


전에도 얘기를 했지만 화이트 스페이스는 띄어쓰기 뿐만 아니라 탭과 엔터를 모두 포함합니다.

```
print("\n 1 \t\n 2 \t\t\n\n 3 4 5 \n 6 \t".split())
```

```
['1', '2', '3', '4', '5', '6']
```



수업을 완료하셨으면 체크해주세요.

 수강생 Q&A 보기

assignment_id=181&op1=%ED%94%84%EB%A1%9C%EA%B7%
assignment_id=181&op1=%ED%94%84%EB%A1%9C%EA%B7%

(/questions/new?

assignment_id=181&op1=%ED%94%84%EB%A1%9C%EA%B7%

< 이전 강의
split (/assignments/147)

다음 강의
코딩에 빠진 닭 > (/assignments/146)