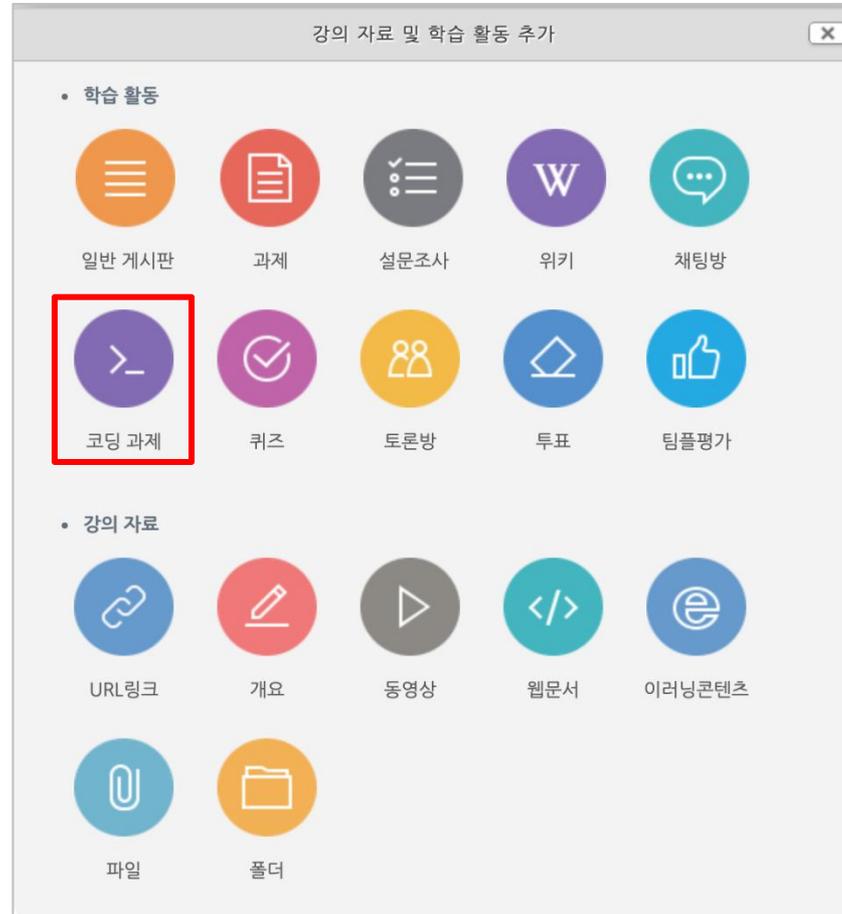


# 코딩 과제

# 1. '강의 자료 및 학습활동 추가' - 코딩 과제



'강의 자료 및 학습활동 추가' - 코딩 과제 선택

## 2. 코딩 과제 설정 - 제목 입력 및 제출 기간 설정

▼ 기본

제목\* Python 예제 : 정규표현식 사용하기

상세 설명

먼저 파이썬에서 정규표현식을 사용하기 위해서는 `Regex`를 위한 모듈인 `re` 모듈을 사용한다.  
`re` 모듈의 `compile` 함수는 정규식 패턴을 입력으로 받아들어 정규식 객체를 리턴하는데, 즉 `re.compile`(검색할문자열) 와 같이 함수를 호출하면 정규식 객체 (`re.RegexObject` 클래스 객체)를 리턴하게 된다.  
`re.RegexObject` 클래스는 여러 메서드들을 가지고 있는데, 이 중 여기서는 특정 문자열을 검색하여 처음 맞는 문자열을 리턴하는 `search()` 메서드를 사용해 본다. 이 `search()` 는 처음 매칭되는 문자열만 리턴하는데, 매칭되는 모든 경우를 리턴하려면 `findall()` 을 사용한다. `search()`는 검색 대상이 있으면 결과를 갖는 `MatchObject` 객체를 리턴하고, 맞는 문자열이 없으면 `None` 을 리턴한다.

메인 화면에 설명 보이기

과제에 대한 안내문 입력

▼ 제출 기간

제출 시작일\* 2019 4월 2 01 23  활성화

제출 마감일 2019 4월 10 09 00  활성화

[뒤 보기](#)

과제 제출 기간 설정

코딩 과제 제목과 설명을 입력하고,  
제출 기간 탭에서 제출 시작/마감일을 설정합니다.

## 2. 코딩 과제 설정 - 제출 조건 설정

▼ 제출 조건

업로드 가능한 파일 수	<input type="text" value="1"/>	<ul style="list-style-type: none"> <li>•업로드 가능한 파일 수 : 실행은 단일 파일만 테스트 가능</li> <li>•제출 방식 : 개인 / 팀 설정</li> <li>•테스트 여부 : 웹 상에서 테스트 후 과제 제출이 가능하도록 허용</li> </ul>
제출 방식	<input type="text" value="개인"/>	
외부 파일 업로드, 코드 붙여넣기 제한*	<input type="text" value="아니오"/>	
테스트 여부*	<input type="text" value="아니오"/>	
업로드 파일 크기*	<input type="text" value="선택"/>	
비밀번호*	<input type="text"/>	
	<input type="checkbox"/> 암호보임	

과제 제출 시 코딩 결과 파일을 업로드 하여 실행 결과를 테스트할 수 있습니다.

### 3. 과제 세부 설정 - Test cases (1)

**강의실 홈** -

**강의정보** ▲

- 참여자목록

**성적/출석관리** ▲

- 학습이력현황
- 학습진도현황
- 오프라인출석부
- 성적부

수강생 알림 ▼

기타 관리 ▼

학생화면

**학습활동** +

**고급 설정** -

+
\*
-

- ▼ 코딩 과제 설정
  - 설정
  - Test cases
  - 실행 옵션
  - Requested files
- ▶ 고급설정
- ▶ 테스트
- 코딩 과제

🏠 > test2018 > Python 예제 : 정규표현식 사용하기

개요

과제 평가

테스트

## Python 예제 : 정규표현식 사용하기

제출 마감일: 2019-04-10 09:00  
 업로드 가능한 파일 수: 1  
 제출 방식: 개인  
 성적 설정: 최고성적: 100  
 실행: 아니오. 자동 평가: 아니오

먼저 파이썬에서 정규표현식을 사용하기 위해서는 Regex를 위한 모듈인 re 모듈을 사용한다.

re 모듈의 compile 함수는 정규식 패턴을 입력으로 받아들여 정규식 객체를 리턴하는데, 즉 re.compile(검색할문자열) 와 같이 함수를 호출하면 정규식 객체 (re.RegexObject 클래스 객체)를 리턴하게 된다.

re.RegexObject 클래스는 여러 메서드들을 가지고 있는데, 이 중 여기서는 특정 문자열을 검색하여 처음 맞는 문자열을 리턴하는 search() 메서드를 사용해 본다. 이 search() 는 처음 매칭되는 문자열만 리턴하는데, 매칭되는 모든 경우를 리턴하려면 findall() 을 사용한다. search()는 검색 대상이 있으면 결과를 갖는 MatchObject 객체를 리턴하고, 맞는 문자열이 없으면 None 을 리턴한다.

VPL

### 3. 과제 세부 설정 - Test cases (2)

> Test cases: Python 예제 : 정규표현식 사용하기 ⓘ

+ D ↗ ?

vpl\_evaluate.cases ▾

```

1 input = 문의사항이 있으면 033-250-7161 으로 연락주시기 바랍니다.
2 output = 033 250-7161
            
```

입력받는 텍스트(input)과 결과출력 텍스트(output) 을 미리 설정하여 학생들이 과제 제출/테스트 시 input/output 결과에 따라 자동 채점이 가능하도록 합니다.

### 3. 과제 세부 설정 - Test cases (3)

**강의실 홈**

강의정보 ▲

- 참여자목록

성적/출석관리 ▲

- 학습이력현황
- 학습진도현황
- 오프라인출석부
- 성적부

수강생 알림 ▼

기타 관리 ▼

학생화면

학습활동 +

고급 설정

- ▼ 코딩 과제 설정
  - 설정
  - **Test cases**
  - 실행 옵션
  - Requested files
- ▶ 고급설정
- ▶ 테스트
- 코딩 과제

test2018 > Python 예제 : 정규표현식 사용하기

**실행 옵션: Python 예제 : 정규표현식 사용하기**

실행 옵션

기준	선택	<ul style="list-style-type: none"> <li>•언어 선택 : 다양한 코딩 언어 지원 (C, C++, C#, JAVA, PHP, PYTHON 등)</li> <li>•실행 : 과제 제출 시 학생이 작성한 코딩 내용을 실행하여 결과를 확인해볼 수 있음</li> <li>•디버그 : 과제 실행 시 디버깅 모드 제공</li> <li>•자동 평가 : test cases 에 의해 설정된 input/output 값으로 자동 평가(채점)</li> </ul>
언어 선택	PYTHON-3: Using python3 with the first fil	
디버그 스크립트	자동 감지	
실행	아니오	
디버그	아니오	
자동 평가	아니오	
제출 시 평가여부	아니오	
자동 채점	아니오	
	옵션 저장	

코딩 과제에 대한 세부 옵션을 설정합니다.

## 4. 코딩 테스트

The screenshot shows a web interface for a coding test. At the top, there are navigation tabs: '개요' (Overview), '과제 평가' (Assignment Evaluation), and '테스트' (Test), with '테스트' highlighted in blue and enclosed in a red box. Below these are buttons for '제출' (Submit), '편집' (Edit), and '코딩 결과' (Coding Results). A toolbar with various icons is also visible, with a red box around the 'Run' icon (a play button) and a red arrow pointing to the console window.

The code editor shows the following Python code:

```

1 import re
2
3 text = "문의 사항이 있으면 033-250-7161 으로 연락주시기 바랍니다."
4
5 regex = re.compile(r'(?P<area>\d{3})-(?P<num>\d{3}-\d{4})')
6 matchobj = regex.search(text)
7 areaCode = matchobj.group("area")
8 num = matchobj.group("num")
9 print(areaCode, num) # 033 250-7161
10

```

The console window displays the output: `033 250-7161`. The console title is 'Console: connection closed (실행중)'. To the right of the console, there is a '개요' (Overview) section with text explaining the use of regular expressions (Regex) and the `re` module.

정규표현식은 Regex를 위한 모듈을 사용한다.

`re` 함수는 정규표현식으로 받아들여 정하는 데, 즉 할문자열)와 하면 정규식 객체 클래스 객체 객체를 리턴한다.

클래스는 여러 개 있는데, 이 문자열을 검색 문자열을 리턴하도록 사용해 본 는 처음 매칭되 하는데, 매칭되 리턴하려 하면 한다. `search()` 있으면 결과를 `Match` 객체를 리턴 없이 없으면 다.

설정된 옵션이 제대로 동작하는지 미리 테스트해볼 수 있습니다.

## 4. 코딩 테스트

개요	과제 평가	<b>테스트</b>		
제출	편집	<b>코딩 결과</b>	성적	이전 제출 목록

최종 제출일 : 2019-04-02 01:48 (다운로드)

### 자동 평가[-]

자동 채점 결과 : 0 / 100

**설명[-]**  
 [-]Test 1  
 Incorrect program output  
 --- Input ---

문의사항이 있으면 033-250-7161 으로 연락주시기 바랍니다.

--- Program output ---

033 250-7141

--- Expected output (text)---

033 250-7161

**[-]Summary of tests**

```
+-----+
| 1 test run/ 0 tests passed |
+-----+
```

test.py

```
1 import re
2
3 text = "문의 사항이 있으면 033-250-7141 으로 연락주시기 바랍니다."
4
5 regex = re.compile(r'(?P<area>Wd(3))-?(?P<num>Wd(3)-Wd(4))')
6 matchobj = regex.search(text)
7 areaCode = matchobj.group("area")
8 num = matchobj.group("num")
9 print(areaCode, num)
```

입력한 과제물이 Test Cases 와 일치하는지 결과를 즉시 확인할 수 있습니다.

## 5. 과제 평가

개요 **과제 평가** 테스트

제출물 선택 과제 제출한 학생  
자동 평가 선택...

제출 횟수 : 과제 제출까지 편집한 횟수를 의미하며 과정을 모니터링 할 수 있습니다.

	이름 / 영문명	최종 제출일	제출 횟수	성적	채점자	채점일
1	김학생	2019-04-02 01:45	2	성적 없음		
2	이학생	2019-04-02 01:54	1	성적 없음		

평가결과 제출물 다운로드 모든 제출물 일괄 다운로드 **최종 제출일을 클릭하면 상세 제출 내역을 확인할 수 있습니다.** VPL

개요 과제 평가 **사용자 김학생**

제출 편집 **코딩 결과** 성적 이전 제출 목록

최종 제출일 : 2019-04-02 01:45 (다운로드)

설명  
교수님 저는 처음이라 잘 모르겠네요.

**과제 제출 내용**

```
test.py
1 import re
2
3 text = "문의 사항이 있으면 033-250-7161 으로 연락주시기 바랍니다."
4
5 regex = re.compile(r'(?P<area>\w{3})-(?P<num>\w{3}-\w{4})')
6 matchobj = regex.search(text)
7 areaCode = matchobj.group("area")
8 num = matchobj.group("num")
9 print(areaCode, num) # 033 250-7161
```

개요 과제 평가 **사용자 김학생**

제출 **편집** 코딩 결과 성적 이전 제출 목록

test.py

```
1 import re
2
3 text = "문의 사항이 있으면 033-250-7161 으로 연락주시기 바랍니다."
4
5 regex = re.compile(r'(?P<area>\d{3})-(?P<num>\d{3}-\d{4})')
6 matchobj = regex.search(text)
7 areaCode = matchobj.group("area")
8 num = matchobj.group("num")
9 print(areaCode, num) # 033 250-7161
```

**과제 제출 내용 실행 / 디버그 / 자동 평가**

## 5. 과제 평가 (자동채점)

개요 **과제 평가** 테스트

제출물 선택 과제 제출한 학생

자동 평가 선택...  
평가되지 않은 학생  
채점되지 않은 학생  
모두

**개인 단위 채점이 아니라 학생들이 제출한 과제 전체를 자동 채점을 할 수 있습니다.**

	이름 / 영문명	최종 제출일	제출 횟수	성적	채점자	채점일
1	김학생	2019-04-02 01:45	2	성적 없음		
2	이학생	2019-04-02 01:54	1	성적 없음		

평가결과 제출물 다운로드 모든 제출물 일괄 다운로드 VPL

성적 100 성적 채점 & 다음 채점결과 삭제

복사 성적 계산

설명

```
-Summary of tests
>-----+
> | 1 test run/ 1 test passed |
>-----+
```

최종 제출일: 2019-04-02 01:45 (다운로드)

	성적	채점자	채점일
1	자동 채점 결과 : 100 / 100		
2	자동 채점 결과 : 100 / 100		

설명

교수님 저는 처음이라 잘 모르겠네요.

자동 평가[-] **채점 결과를 확인 후 추가 코멘트를 할 수 있고 재채점을 할 수 있습니다.**

자동 채점 결과 : 100 / 100

설명[-]

```
-Summary of tests
>-----+
> | 1 test run/ 1 test passed |
>-----+
```

test.py

```
1 import re
2
3 text = "문위사 함이 있으면 033-250-7161 으로 연락주시기 바랍니다."
4
5 regex = re.compile(r'(P<area>Wd(3))-?(P<num>Wd(3)-Wd(4))')
6 matchobj = regex.search(text)
7 areaCode = matchobj.group("area")
8 num = matchobj.group("num")
9 print(areaCode, num) # 033 250-7161
```



