

COSE212 – Programming Language

실습환경 안내

TryML

TryML

- OCaml 실행환경 in Web
- 과제 수행 및 제출에 활용될 홈페이지
 - <https://tryml.korea.ac.kr>

TryML



Assignment Policy

- Discussion with fellow students is encouraged including how to approach the problem. **However, your code must be your own.**
- Violating above rule gets you **0 points** for the entire HW score.
- See [README.ml](#) for detailed information.

Homework

Feedback

Option

Run

Submit

README.ml

```
1 (*****README*****  
2  
3 1. Before starting programming, you should login and choose the problem  
4   located beside.  
5  
6 2. Check the assignment policy again (MUST NOT COPY THE HW).  
7  
8 3. We will grade with the final version of your code. Do not forget to  
9   click "submit button" (or press ctrl + S key) after programming.  
10  (Since you can submit unlimited times before due date, you can use it  
11  as saving your code.)  
12  
13 4. The editor will show the final submitted implementation. When you take  
14  a rest or stop the programming, click the "submit button" (or press  
15  ctrl + S key) for saving your task.  
16  
17 5. You can do your homework with the provided template. Fill the "TODO" part  
18  of each problem.  
19
```

Output

Press "Run" (ctrl + enter) to execute your code.

TryML

- 과제 채점에 이용되므로 본인의 학번을 정확하게 입력하여 가입
- **비밀번호를 잊어버리지 않도록 주의**

COSE212 - Programming Language Sign Up Login

SIGN UP [X]

USERNAME
heojunyong

STUDENT ID
20XXXXXXXX

E-MAIL
heo1569@korea.ac.kr

PASSWORD
.....

CONFIRM
.....

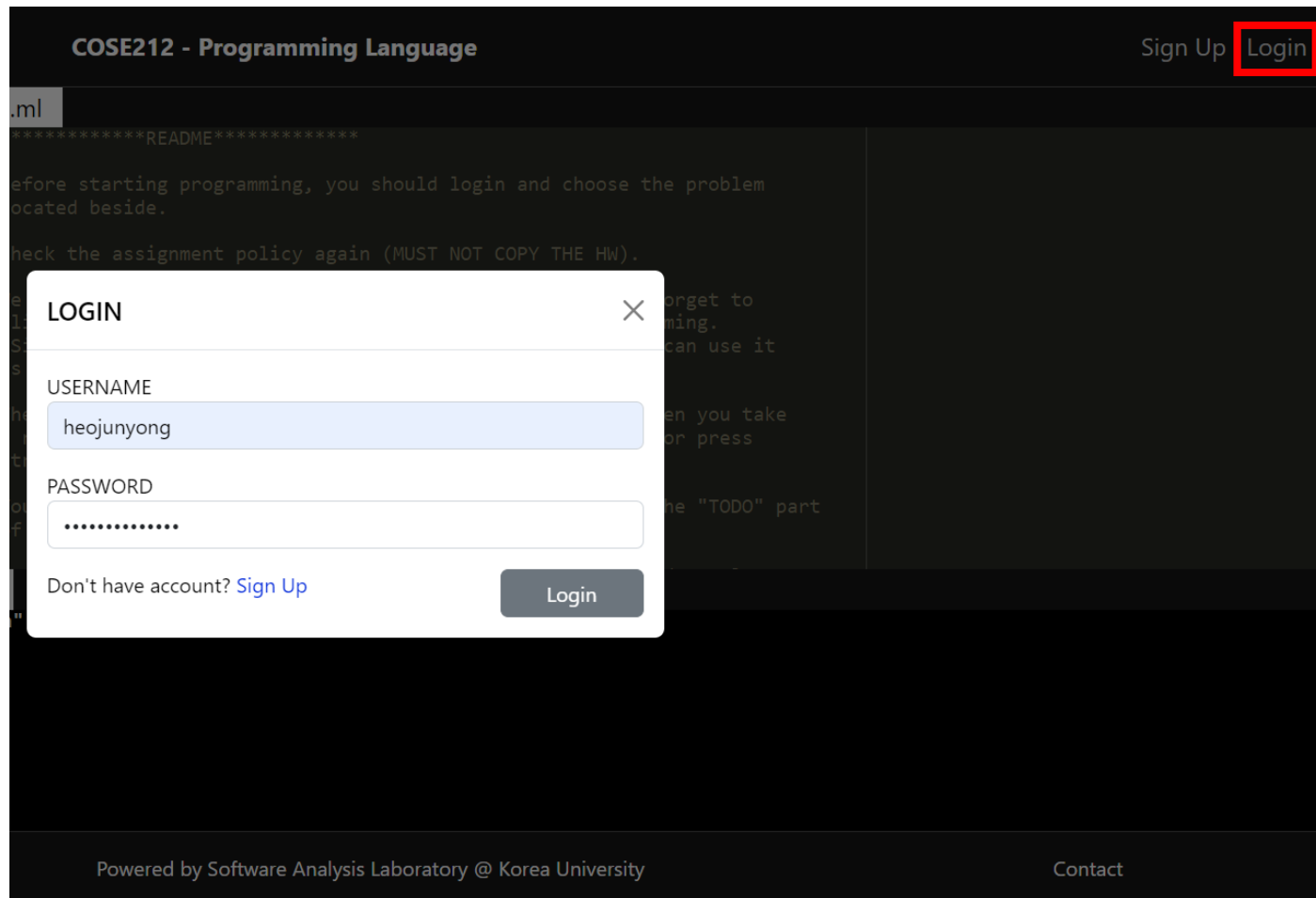
I agree with the [Privacy Policy](#)

Sign Up

Powered by Software Analysis Laboratory @ Korea University Contact

TryML

- 가입한 아이디 및 비밀번호로 로그인



TryML

- Assignment Policy 및 README.ml 내용 숙지 필수

COSE212 - Programming Language

heojunyong Logout

Assignment Policy

- Discussion with fellow students is encouraged including how to approach the problem. **However, your code must be your own.**
- Violating above rule gets you **0 points** for the entire HW score.
- See **README.ml** for detailed information.

Homework

Feedback

Option

README.ml

```
1 (*****README*****  
2  
3 1. Before starting programming, you should login and choose the problem  
4   located beside.  
5  
6 2. Check the assignment policy again (MUST NOT COPY THE HW).  
7  
8 3. We will grade with the final version of your code. Do not forget to  
9   click "submit button" (or press ctrl + S key) after programming.  
10  (Since you can submit unlimited times before due date, you can use it  
11   as saving your code.)  
12  
13 4. The editor will show the final submitted implementation. When you take  
14   a rest or stop the programming, click the "submit button" (or press  
15   ctrl + S key) for saving your task.  
16  
17 5. You can do your homework with the provided template. Fill the "TODO" part  
18   of each problem.  
19
```

Output

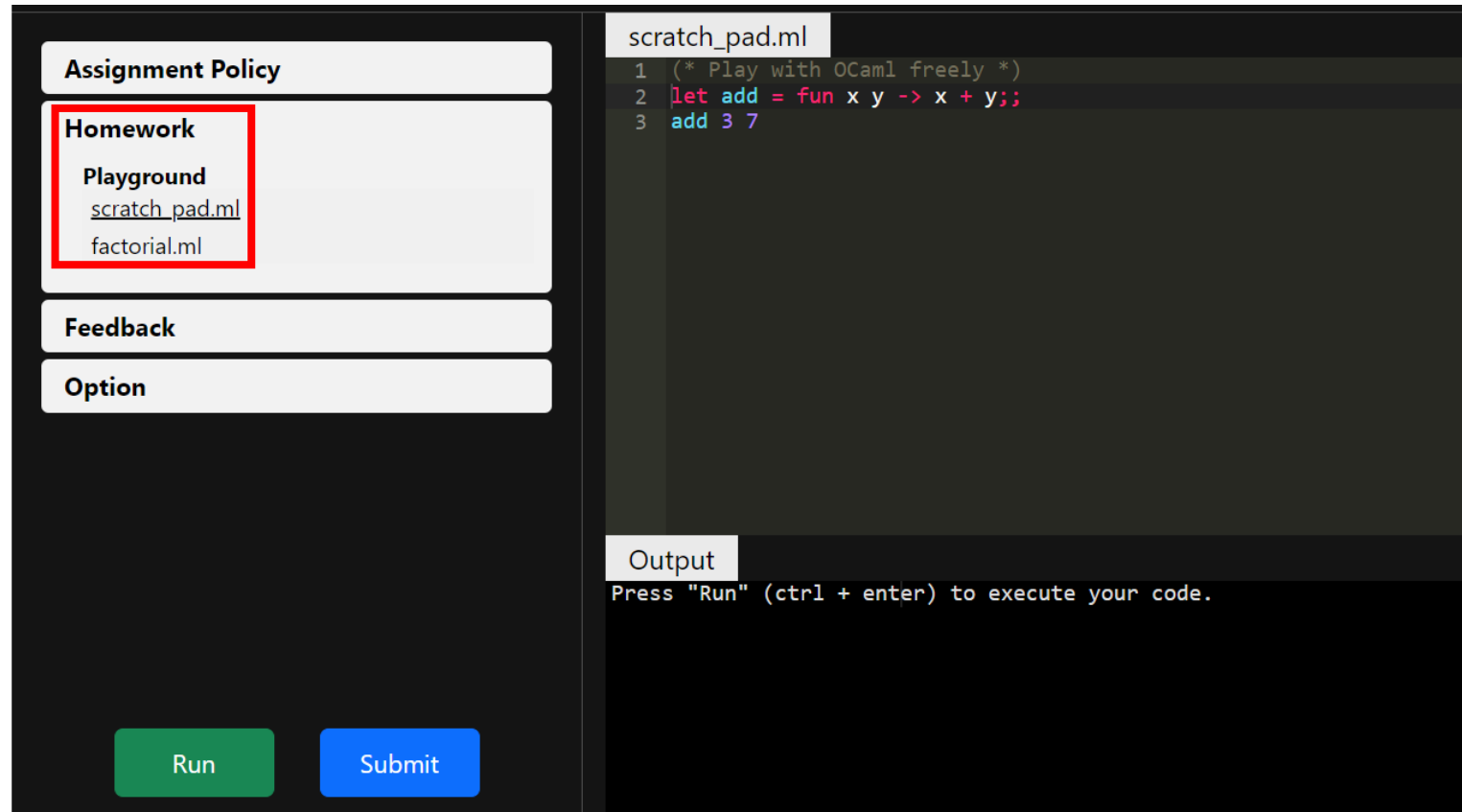
Press "Run" (ctrl + enter) to execute your code.

Run Submit

Privacy Policy Powered by Software Analysis Laboratory @ Korea University Contact

TryML

- `scratch_pad.ml`: 자유롭게 코드를 작성하고 실행할 수 있는 공간
- 과제 관련 탭은 추가 예정



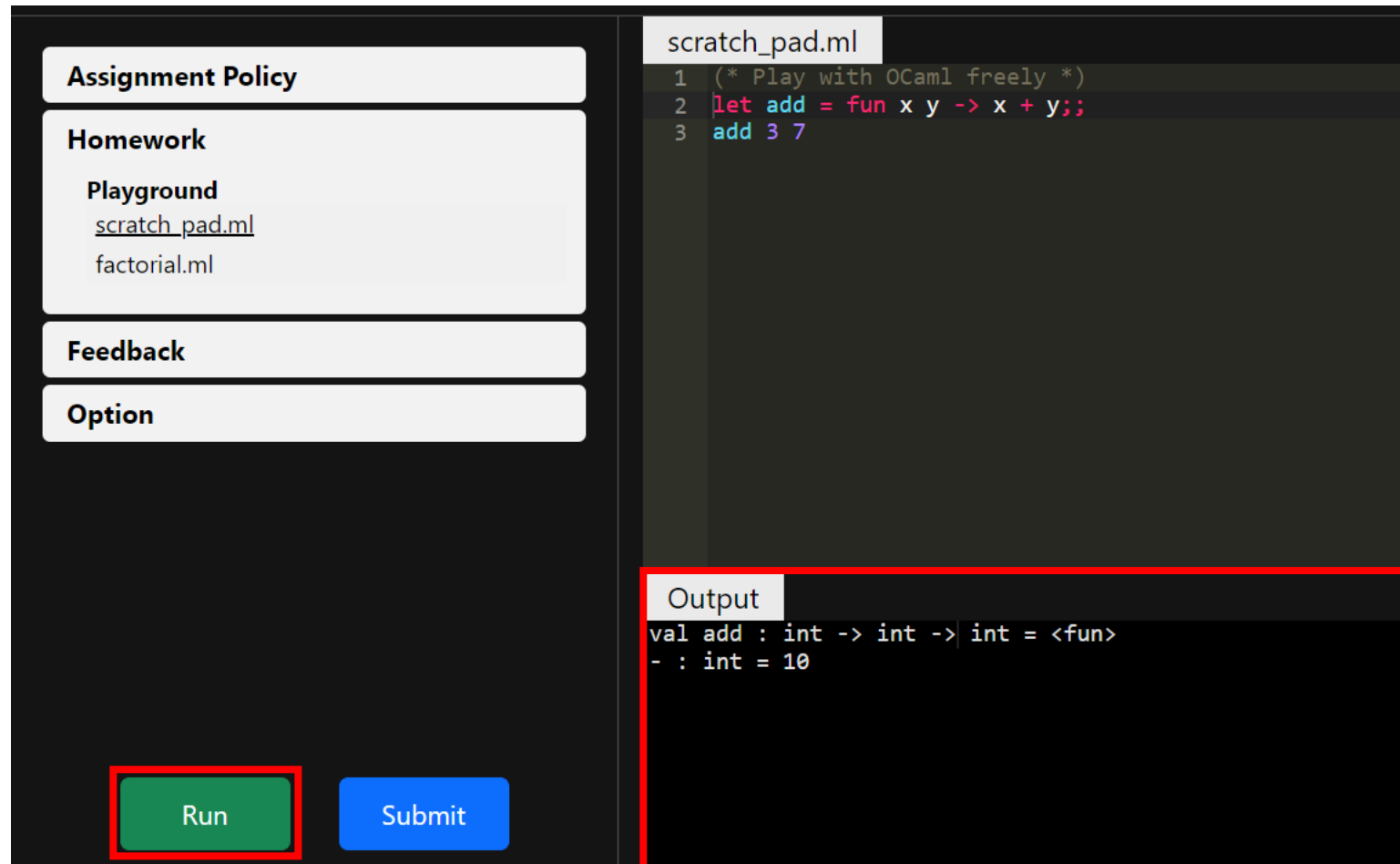
The screenshot displays the TryML interface. On the left, a sidebar contains several tabs: 'Assignment Policy', 'Homework', 'Feedback', and 'Option'. The 'Homework' tab is highlighted with a red border and contains a 'Playground' section with links to `scratch_pad.ml` and `factorial.ml`. The main area is a code editor titled 'scratch_pad.ml' containing the following OCaml code:

```
1 (* Play with OCaml freely *)  
2 let add = fun x y -> x + y;;  
3 add 3 7
```

Below the code editor is an 'Output' section with the instruction: 'Press "Run" (ctrl + enter) to execute your code.' At the bottom of the interface, there are two buttons: a green 'Run' button and a blue 'Submit' button.

TryML

- 실행: **Run** 버튼 클릭 또는 Ctrl + Enter



The screenshot displays the TryML interface. On the left, there is a sidebar with sections: "Assignment Policy", "Homework" (containing links for "Playground", "scratch_pad.ml", and "factorial.ml"), "Feedback", and "Option". At the bottom of the sidebar, there are two buttons: "Run" (highlighted with a red box) and "Submit". The main area on the right is a code editor titled "scratch_pad.ml" containing the following OCaml code:

```
1 (* Play with OCaml freely *)  
2 let add = fun x y -> x + y;;  
3 add 3 7
```

Below the code editor is an "Output" window (also highlighted with a red box) showing the execution results:

```
val add : int -> int -> int = <fun>  
- : int = 10
```


TryML

- 저장: Ctrl + S

```
scratch_pad.ml  
1 (* Play with OCaml freely *)  
2 let add = fun x y -> x + y;;  
3 add 3 7|
```

↓ 코드 수정

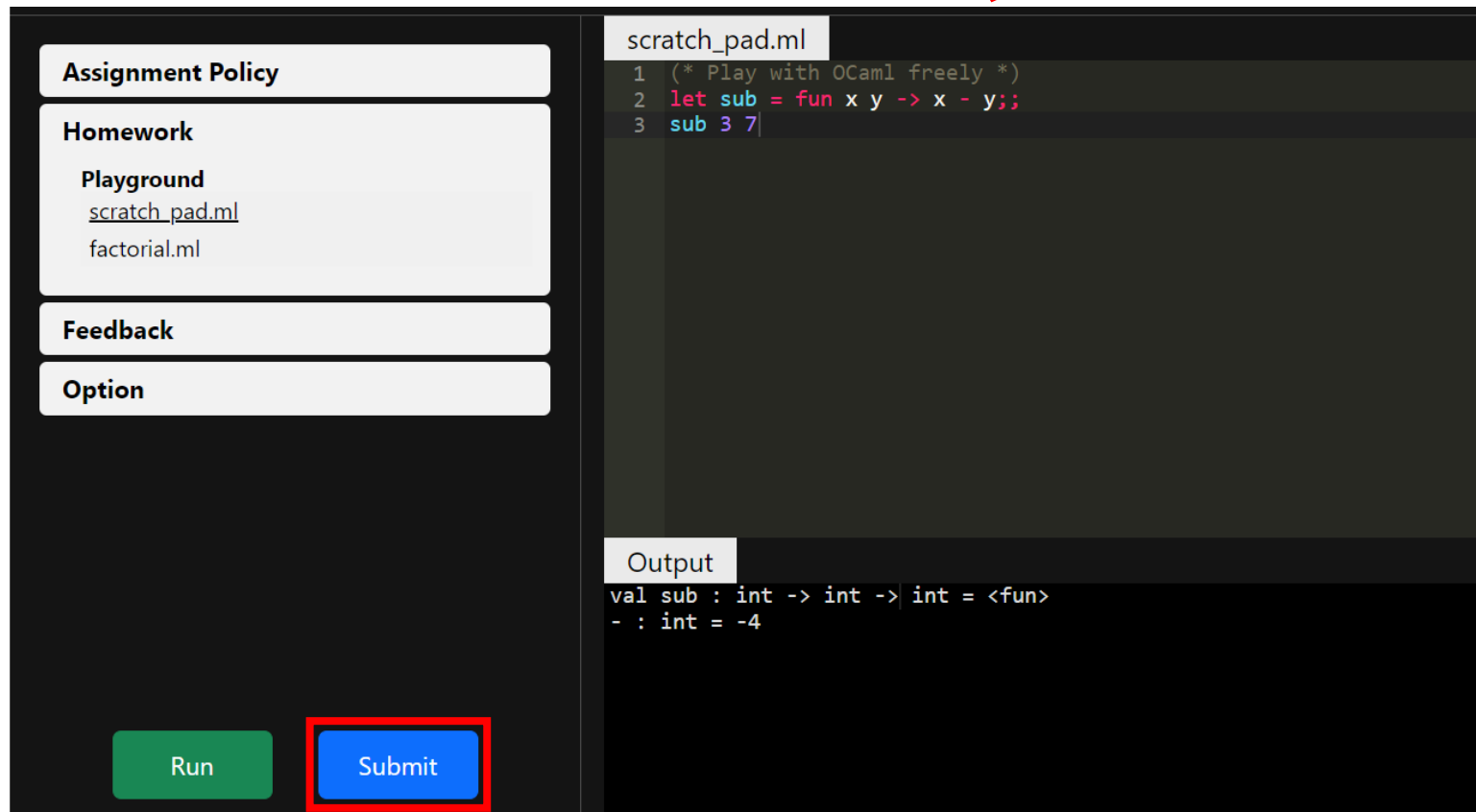
```
scratch_pad.ml|*  
1 (* Play with OCaml freely *)  
2 let sub = fun x y -> x - y;;  
3 sub 3 7|
```

↓ Ctrl + S

```
scratch_pad.ml  
1 (* Play with OCaml freely *)  
2 let sub = fun x y -> x - y;;  
3 sub 3 7|
```

TryML

- 제출: **Submit** 버튼 클릭 (**Run** 버튼으로 제출 안됨)
- 과제 기한이 지나면 제출되지 않으니 주의 (지각 제출 허용 안함)



The screenshot displays the TryML interface. On the left, there is a sidebar with sections: "Assignment Policy", "Homework" (containing links for "Playground", "scratch_pad.ml", and "factorial.ml"), "Feedback", and "Option". The main area is a code editor titled "scratch_pad.ml" containing the following OCaml code:

```
1 (* Play with OCaml freely *)  
2 let sub = fun x y -> x - y;;  
3 sub 3 7
```

Below the code editor is an "Output" section showing the result of the code execution:

```
val sub : int -> int -> int = <fun>  
- : int = -4
```

At the bottom of the interface, there are two buttons: a green "Run" button and a blue "Submit" button. The "Submit" button is highlighted with a red rectangular border.

TryML. 라이브러리 사용 방지

- 과제 수행에 있어 라이브러리 사용을 금지함
- 이에 따라 모듈 접근을 원천적으로 방지하기 위해,
1) 마침표를 포함한 코드는 실행되지 않음

```
forall.ml
1  exception NotImplemented;;
2
3  let forall: ('a -> bool) -> 'a list -> bool
4  =fun f l -> List.for_all f l;;
```

Output

```
Your code is not allowed to execute.
```

TryML. 라이브러리 사용 방지

- 과제 수행에 있어 라이브러리 사용을 금지함
- 2) open 키워드를 사용하는 코드는 실행되지 않음

```
forall.ml
1  exception NotImplemented;;
2
3  open List;;
4
5  let forall: ('a -> bool) -> 'a list -> bool
6  =fun f l -> for_all f l;;
```

Output

```
Your code is not allowed to execute.
```

기타

- 과제 관련 질문은 [블랙보드 → 토론실 → 강의 Q&A]에 올려 주시면 됩니다.
- 이외의 궁금한 사항은 조교에게 문의하시면 됩니다.
 - 허준용 (heo1569@korea.ac.kr)
 - 신지호 (tswlgh0801@gmail.com)