

OurScheme

PROJECT 1

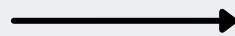


Outline

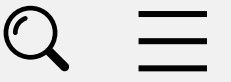
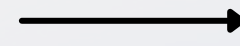
系統操作

PROJECT

測試工具

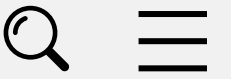


系統操作



安裝Discord

<https://discord.com>

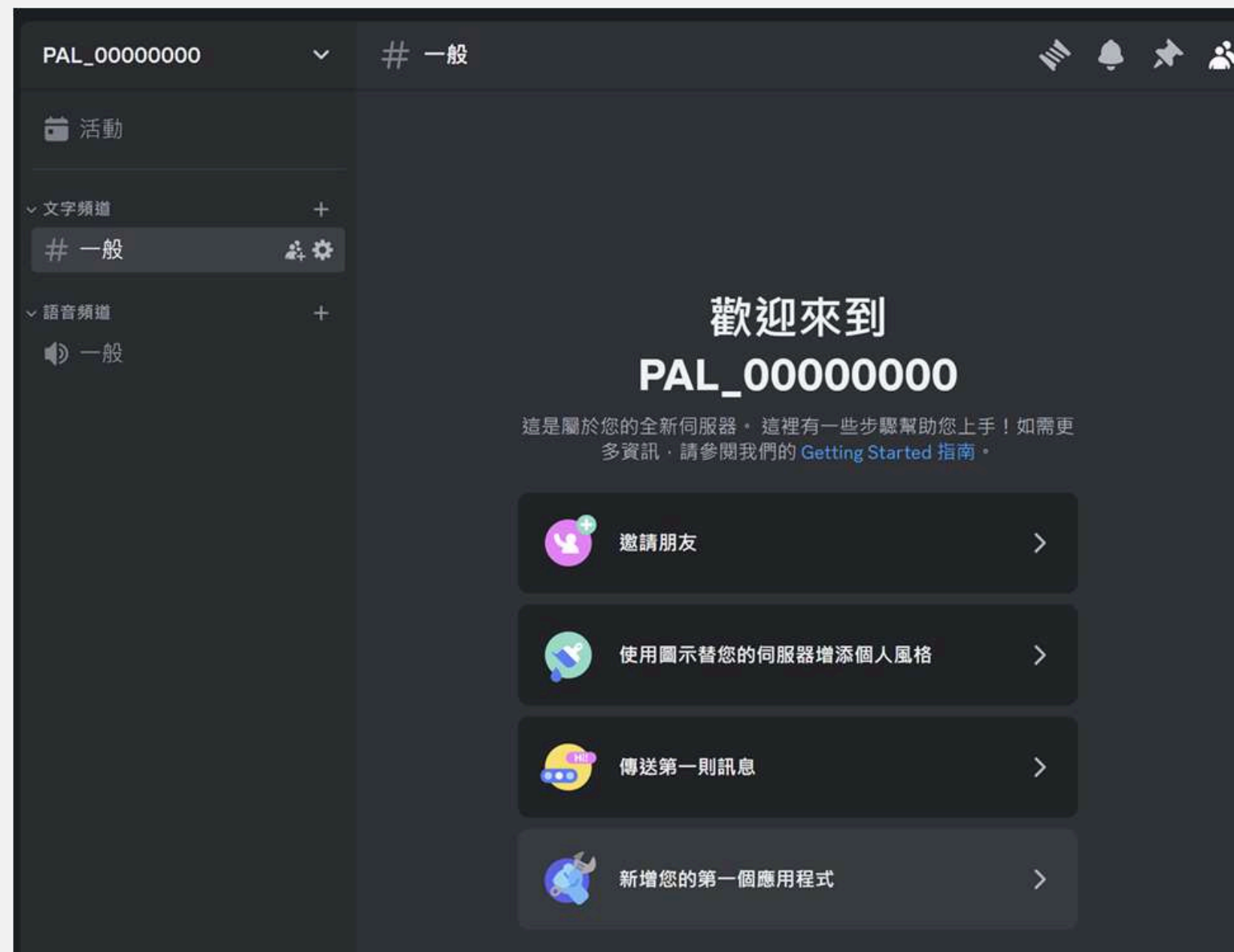


Step1:

在Discord建立帳號並登入

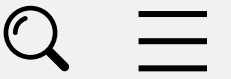
<https://discord.com>

Step2: 新增伺服器



介面最左邊可以新增伺服器，將伺服器名稱設定為PAL_你的學號

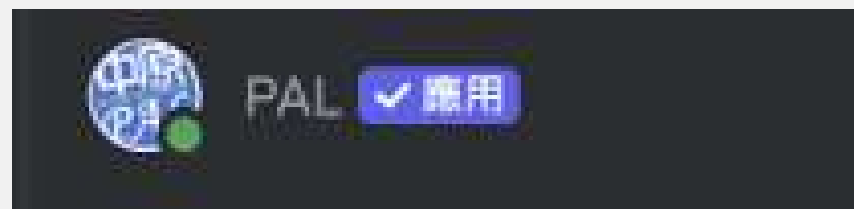
Step3: 將PAL機器人加入伺服器



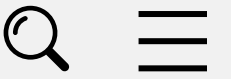
新增機器人至你的伺服器，選擇剛才建立的伺服器，點選繼續、授權。

https://discord.com/oauth2/authorize?client_id=1265725772094767149&permissions=8&integration_type=0&scope=bot

完成後會右邊會有PAL機器人出現



Step4: 建立文字頻道



- 在建立好的discord伺服器中建立一個文字頻道，頻道名稱:上傳
- 切勿勾選 **私人頻道**，機器人僅會回復此文字頻道的訊息。



The image shows the 'Create Channel' dialog box in Discord. At the top, it says '建立頻道' (Create Channel) and '在文字頻道中' (In text channels). Below this is the '頻道類別' (Channel Type) section, which has two options: '文字' (Text) with a radio button that is selected, and '語音' (Voice) with an unselected radio button. The '文字' option has a description: '傳送訊息、圖片、GIF、表情符號、意見和雙關語'. The '語音' option has a description: '透過語音通話、視訊通話和畫面分享來相聚交流'. Below this is the '頻道名稱' (Channel Name) section, which has a text input field containing '# 上傳'. At the bottom, there is a toggle switch for '私人頻道' (Private Channel), which is currently turned off. Below the toggle, it says '只有挑選出的成員及身分能檢視這個頻道。' (Only selected members and roles can view this channel). At the very bottom, there are two buttons: '取消' (Cancel) and '建立頻道' (Create Channel).

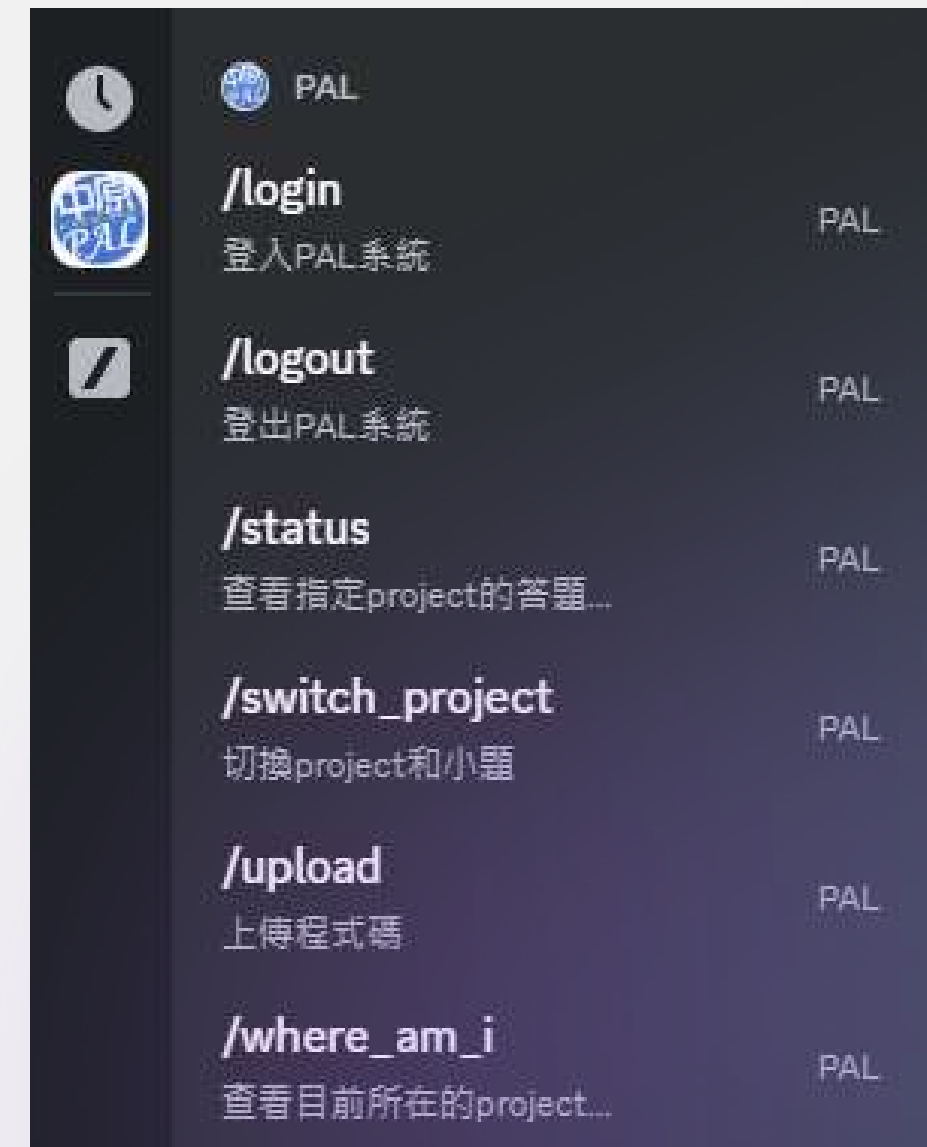
Step5: 功能操作

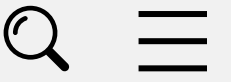
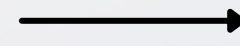
/login

student_id (必填): 你的學號

password (必填): 你的密碼

**若無法登入/登入失敗/不明原因錯誤
請盡快聯繫助教!!!**





剩餘部分請至說明文件詳閱，

有問題請聯繫助教

助教信箱：cycuplassistant@gmail.com

Project 1

- 互動式直譯器
- 介紹 OurScheme
- 抽象語法樹 (Abstract Syntax Tree)
- 格式化輸出 (Pretty Print)
- 處理錯誤

互動式直譯器？

REPL (Read-Eval-Print Loop)

- Read：讀取使用者輸入的指令
- Eval：解析並執行該指令
- Print：輸出執行結果
- Loop：等待下一個輸入，形成迴圈

Welcome to OurScheme!

> (1 . (2 . 3))

)

(1

2

.

3

)

> (exit)

Thanks for using OurScheme!

```
# Step 1: 印出歡迎訊息
Print "Welcome to OurScheme!"
Print "\n"
Print "> " # 提示使用者輸入
```

```
# Step 2: 進入 REPL 迴圈
```

```
Repeat:
```

```
    # 讀取使用者輸入的 S-Expression
```

```
    expr = ReadSExp()
```

```
    # 如果讀取到 EOF (檔案結尾)，跳出迴圈
```

```
    If expr == END-OF-FILE:
```

```
        Print "ERROR (no more input) : END-OF-FILE encountered"
```

```
        Break
```

```
    # 若使用者輸入的是 '(exit)'，結束迴圈
```

```
    If expr == "(exit)":
```

```
        Break
```

```
    # 格式化並輸出 S-Expression
```

```
    PrintSExp(expr)
```

```
    # 顯示新的提示符號，等待下一次輸入
```

```
    Print "> "
```

```
# Step 3: 結束程式
```

```
Print "\n"
```

```
Print "Thanks for using OurScheme!"
```

什麼是 **OurScheme** ？

- OurScheme 是基於 Lisp/Scheme 的一種語言
- 語法以 **S-Expression** (S-Exp) 為基礎
- 由原子 (Atoms) 和清單 (Lists) 組成
- 區分大小寫 apple != Apple
- 支援數字、字串、符號 (Symbols)、布林值等

OurScheme tokens

Token 類型	對應字元	備註
LEFT-PAREN	(左括號，代表 list 開始
RIGHT-PAREN)	右括號，代表 list 結束
INT	123, +123, -123	整數型別，允許正負號
STRING	"string's (example)."	字串，必須使用雙引號包圍，不允許跨行
DOT	.	可作為 Dotted Pair 分隔符號
FLOAT	123.567, 123., .567, +123.4, -.123	浮點數，輸出需統一小數點後三位 ("%0.3f")
NIL	nil, #f	代表 `false`，必須統一輸出為 `nil`
T	t, #t	代表 `true`，必須統一輸出為 `#t`
QUOTE	'	單引號，表示 `quote` 簡寫
SYMBOL	abc, a.b, a-B!c? (區分大小寫)	符號（變數名稱），大小寫區分

Syntax of OurScheme

$\langle \text{S-exp} \rangle ::= \langle \text{ATOM} \rangle$

| LEFT-PAREN $\langle \text{S-exp} \rangle$ { $\langle \text{S-exp} \rangle$ } [DOT $\langle \text{S-exp} \rangle$] RIGHT-PAREN

| QUOTE $\langle \text{S-exp} \rangle$

$\langle \text{ATOM} \rangle ::= \text{SYMBOL} \mid \text{INT} \mid \text{FLOAT} \mid \text{STRING} \mid \text{NIL} \mid \text{T} \mid \text{LEFT-PAREN RIGHT-PAREN}$

| : 或

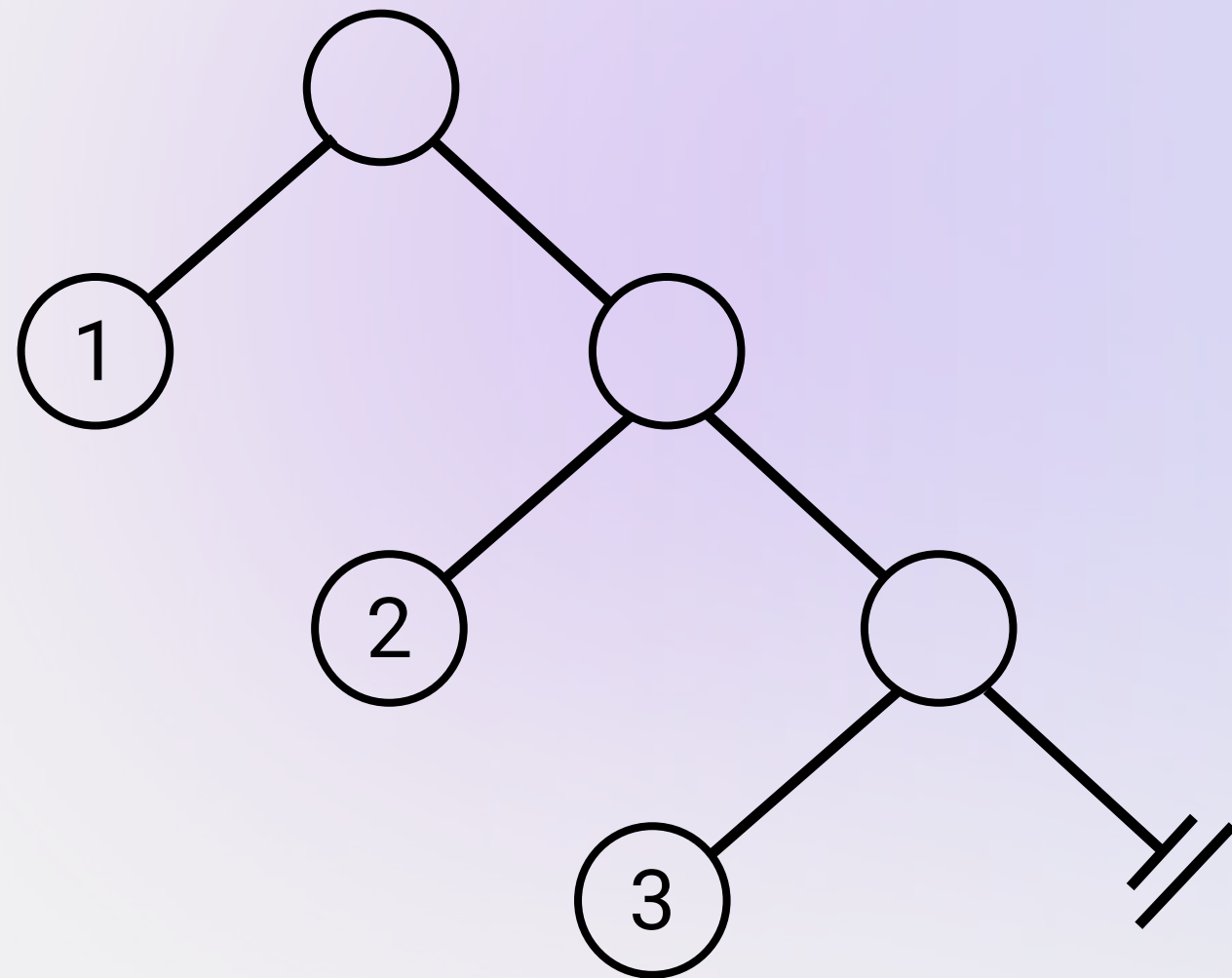
{ } : 出現0次或多次

[] : 出現0次或1次

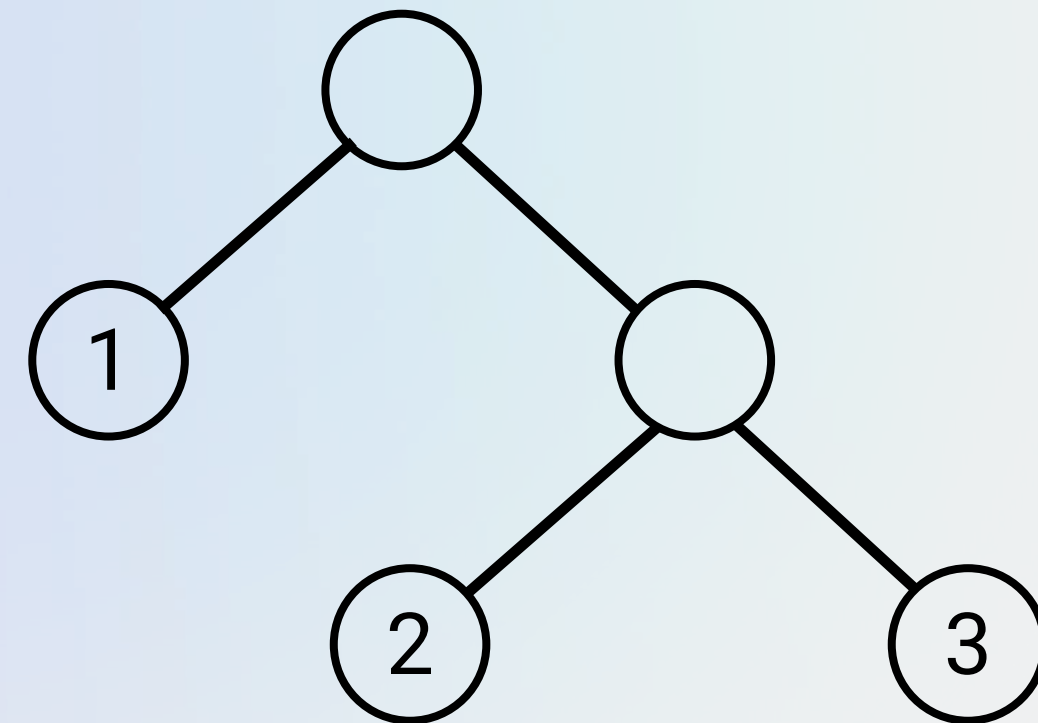
抽象語法樹 AST

AST 是一種樹狀結構，用來表示 OurScheme 的語法結構，並且搭配遞迴下降解析器 (recursive decent parser)使得解析、評估 (Evaluation)、優化 等操作更加容易處理。
例如：

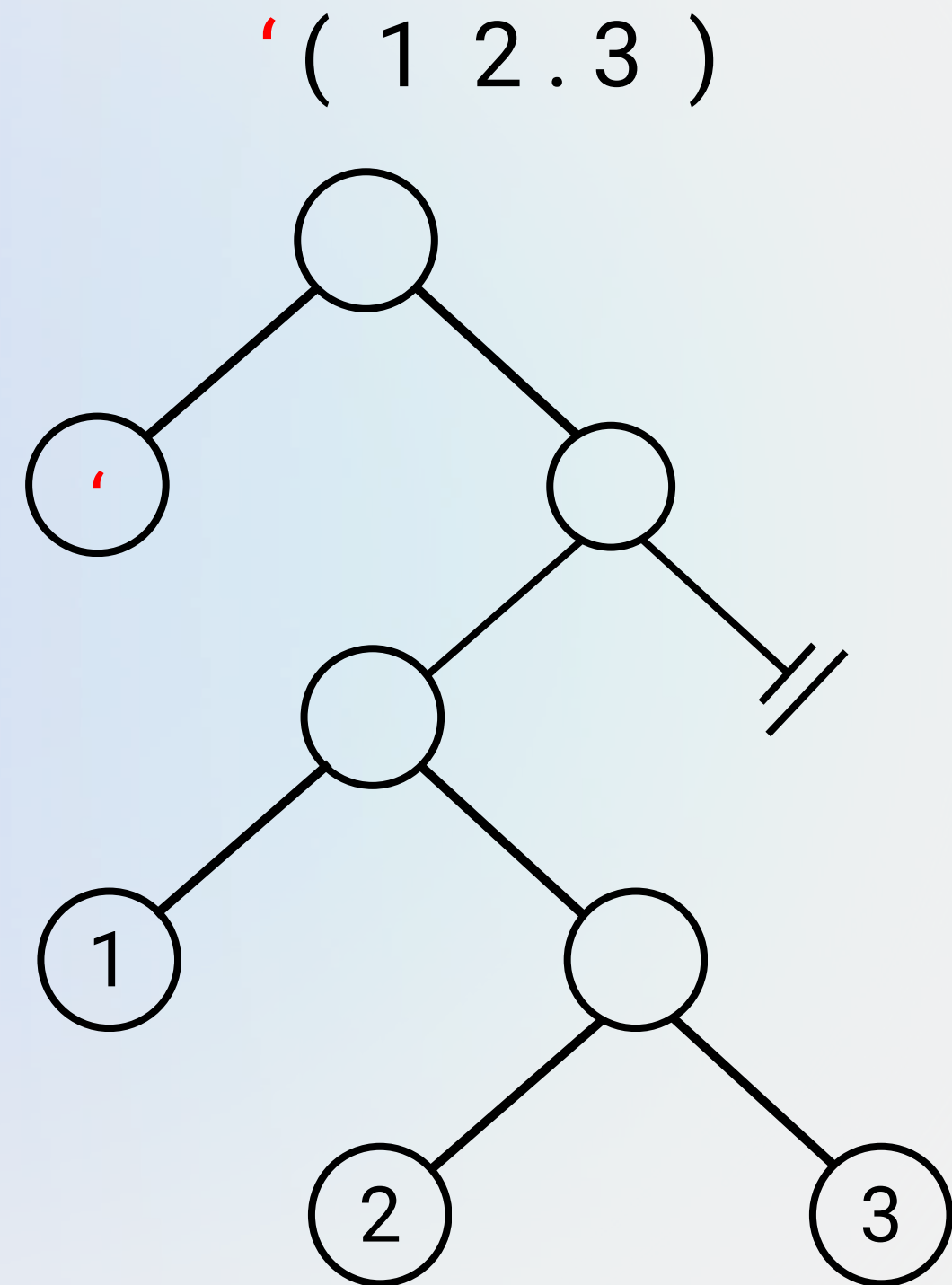
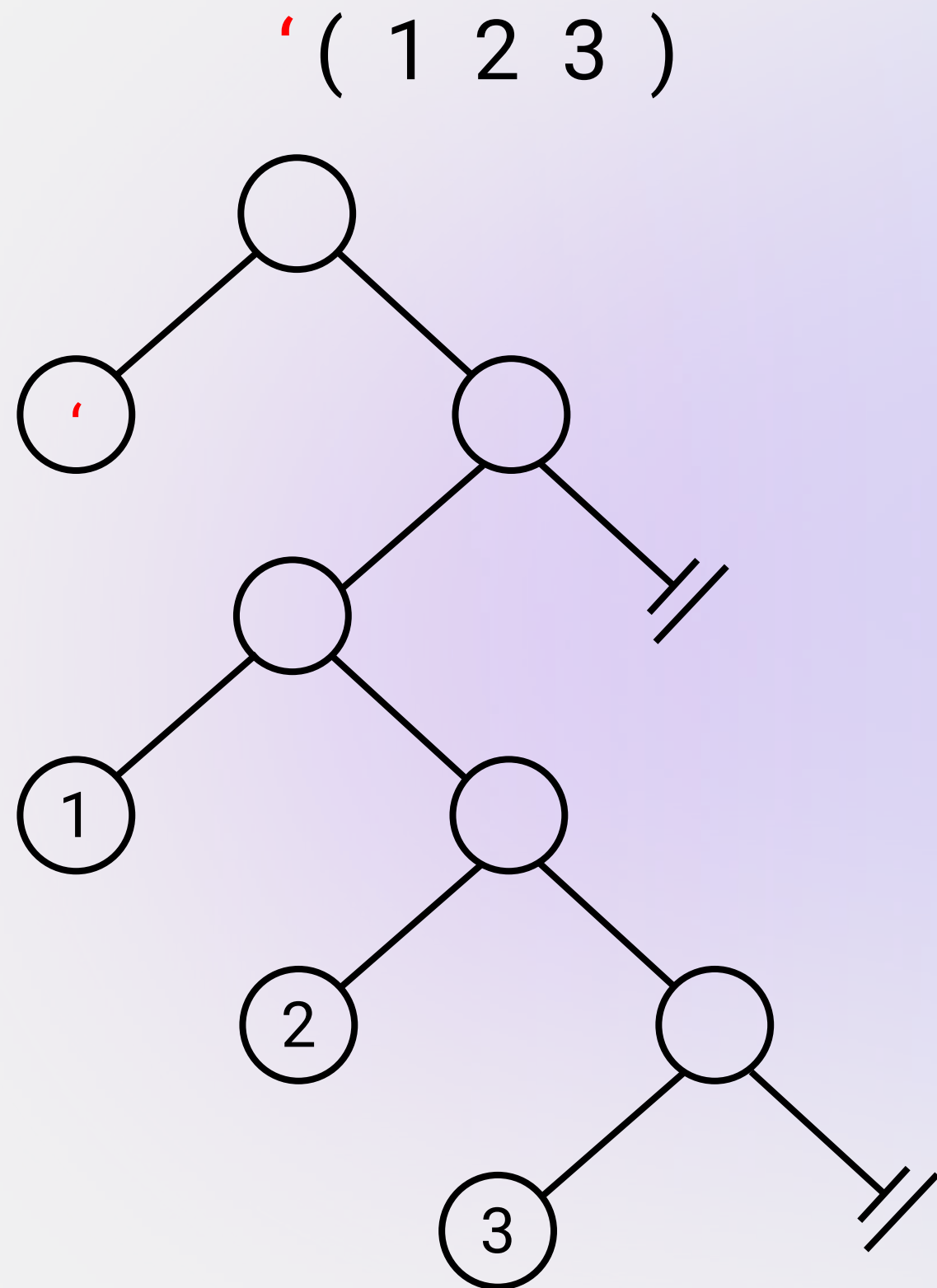
(1 2 3)



(1 2 . 3)



抽象語法樹 AST



格式化輸出 **Pretty print**

(1 2 3)

```
( 1  
 2  
 3  
)
```

(1 2 . 3)

```
( 1  
 2  
 .  
 3  
)
```

' (1 2 . 3)

```
( quote  
 ( 1  
   2  
   .  
   3  
 )  
)
```

處理錯誤

錯誤類型	發生條件	對應錯誤訊息	舉例
未預期的 Token	碰到不符合文法的token (如 . 開頭的錯誤輸入)	ERROR (unexpected token) : atom or '(' expected when token at Line X Column Y is >>...<<	(1 2 .)
缺少右括號)	S-Exp 未完整封閉，遇到 . 或) 時語法錯誤	ERROR (unexpected token) : ')' expected when token at Line X Column Y is >>...<<	(1 2 . 3 4)
未關閉字串	遇到 " 開頭但沒有對應 " 結 束，且換行	ERROR (no closing quote) : END-OF- LINE encountered at Line X Column Y	("hello)
檔案結束 (EOF)	讀取輸入時沒有 (exit)，但 檔案提前結束	ERROR (no more input) : END-OF- FILE encountered	-

處理錯誤

Once the attempt to read in an S-expression fails, the line containing the error-token is ignored.

The system starts to read in an S-expression from the next input line.



一旦嘗試讀取 S-expression 失敗，含有 error-token 的那一行將被忽略。系統會從下一個輸入行開始讀取 S-expression。

Proj1 完成步驟

1. 建立 REPL (Read-Eval-Print-Loop)
 - while 迴圈不斷讀取輸入直到 (exit) 或 EOF。
2. 實作 ReadSExp()
 - 解析 token，構建 AST (Abstract Syntax Tree)。
3. 實作 PrintSExp()
 - 遵循 pretty-print 規則 來輸出 S-Expression。
4. 實作錯誤處理
 - 未預期的 token、括號錯誤、EOF、字串錯誤都要報錯並跳過當行。

PL線上測試系統



歡迎來到 PL 可視化

中原大學資訊工程學系「程式語言」課程學習輔助工具，用於程式碼的可視化和執行追蹤。

OurScheme

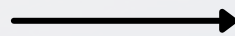
<https://cycu-ice-pl.github.io/website/#/>

自動化測試工具

- 助教本人開發
- 一鍵測試所有自訂測資
- 支援測資匯入/匯出
- 「知識只有在分享時才有價值。」

互相分享測試資料有助於感情增溫





祝各位順利~