

معرفی زبان تسلنگ

مستند حاضر زبان ساده‌ی تسلنگ (TSLANG) را معرفی می‌کند. این نسخه‌ی زبان تسلنگ به زبان PostScript شباهت دارد و بر خلاف نسخه‌های گذشته‌ی آن، یک زبان پشته‌ای است. در گام‌های تمرین عملی درس طراحی کامپایلر، بخش‌هایی از یک مترجم برای این زبان نوشته می‌شوند. قواعد این زبان در ادامه‌ی این مستند بیان می‌شوند.

۱ زبان تسلنگ دارای سه نوع داده‌ی اصلی است: اعداد صحیح، رشته‌ها و دنباله‌ای از دستورات.

۲ رشته‌های در این زبان بین دو علامت «(» و «)» قرار می‌گیرند. مثال «(Hello world!)».

۳ دنباله‌ی دستورات بین علامت‌های «{» و «}» قرار می‌گیرند. یک دنباله از دستورات می‌توانند اجرا شوند یا می‌توان از آنها برای تعریف توابع استفاده نمود.

۴ در زبان تسلنگ، همه‌ی عملوندها روی یک پشته قرار می‌گیرند و عملگرها و توابع روی چند عملوند بالای پشته عمل می‌کنند. برای نمونه، در برنامه‌ی زیر ابتدا یک رشته در بالای پشته قرار می‌گیرد و سپس تابع داخلی `print` آن را چاپ می‌کند.

```
(Hello World!) print
```

۵ در نمونه‌ی زیر، ابتدا عدد دو در پشته قرار می‌گیرد، سپس عدد پنج قرار می‌گیرد و سپس تابع «+» فراخوانی می‌شود. این تابع دو عدد از پشته بر می‌دارد و مجموع آنها را روی پشته قرار می‌دهد.

```
5 2 +
```

۶ در زبان تسلنگ می‌توان یک متغیر یا تابع را با تابع داخلی `def` تعریف کرد. ابتدا نام متغیر پس از علامت «/» در پشته قرار می‌گیرد، سپس مقدار این متغیر و در پایان تابع `def` فراخوانی می‌گردد. در مثال زیر متغیر `lim` با مقدار هفت مقدار دهی می‌شود.

```
/lim 5 2 + def
```

۷ برای تعریف تابع نیز می توان از def استفاده کرد و برای مقدار، دنباله ای از دستورات را مشخص نمود. برای نمونه، تابع sum3 مجموع سه عدد بالای پشته را محاسبه می کند و حاصل را بالای پشته قرار می دهد.

```
/sum3 { + + } def
```

۸ برای فراخوانی این تابع، ابتدا باید سه عدد در پشته قرار داد و سپس آن را فراخوانی نمود.

```
5 2 7 sum3
```

۹ برای عبارتهای شرطی می توان از توابع if و ifelse استفاده نمود. در نمونهی زیر، ابتدا مجموعهی از دستورات (که رشتهی First را چاپ می کند) روی پشته قرار داده می شود، سپس عدد دو و سپس مقدار متغیر lim روی پشته قرار می گیرد. سپس تابع < در پشته قرار می گیرد که دو عدد بالای پشته را بر می دارد، مقایسه می کند و حاصل مقایسه را روی پشته قرار می دهد. سپس تابع if اجرا می شود. این تابع دو مقدار از پشته بر می دارد. اگر مقدار اول غیر صفر باشد، مقدار دوم را اجرا می کند (مقدار دوم باید دنباله ای از دستورات باشد).

```
{ (Smaller) print } lim 5 < if
```

۱۰ تابع ifelse مشابه if است با این تفاوت که سه مقدار را از پشته بر می دارد. در صورتی که مقدار بالای پشته یک باشد، مقدار سوم پشته و در غیر این صورت مقدار دوم پشته را اجرا می کند.

```
{ (Smaller) print } { (Larger) print } lim 5 < ifelse
```

۱۱ جدول زیر توابع داخلی تسلنگ را نشان می دهد.

تابع	توضیح
[A] [B] <code>exch</code>	دو مقدار بالای پشته را جایگزین می کند.
[X] <code>dup</code>	از مقدار بالای پشته یک کپی می گیرد.
[first] [count] <code>roll</code>	دو مقدار را از پشته بر می دارد و سپس به تعداد ورودی اول مقدار در بالای پشته را به مقدار پارامتر دوم می چرخاند.
[count] {op} <code>repeat</code>	دنباله‌ی دستورات بالای پشته را به تعداد پارامتر اول تکرار می کند و آنها را از پشته بر می دارد.
[/name] [value] <code>def</code>	متغیری با نام ورودی اول و با مقدار ورودی دوم تعریف می کند و این دو ورودی را از پشته حذف می کند.
(Message) <code>print</code>	رشته‌ی بالای پشته را چاپ می کند و آن را حذف می نماید.

۱۲ گسترش‌های زیر نیز (که ساختاری مثل دستورات پیش‌پردازشگر در زبان C دارند) در زبان تسلنگ وجود دارند.

```
#def var
  3 2 +
#end
#def op
  print
#end
#ifdef var
  #ifdef op
    var op
  #else
    (op undefined) print
  #end
#else
  (var undefined) print
#end
```

قواعد تجزیه‌ی زبان تسلنگ

در ادامه ساختار BNF زبان تسلنگ نمایش داده شده است. همچنین در برنامه‌های زبان تسلنگ، علامت % و حروفی که بعد از آن آمده‌اند تا آخر خط توضیح (Comment) محسوب می‌شوند.

```
prog ::=      stmts
stmts ::=     stmts stmt |
stmt  ::=     exts |
           { stmts } |
           str |
           num |
           iden |
           oper
exts  ::=     #ifdef iden stmts #end |
           #ifdef iden stmts #else stmts #end |
           #def iden stmts #end |
           #app iden stmts #end
num   ::=     [0-9]+
oper  ::=     (+|-|*|<|<=|>|>=|==)
iden  ::=     [a-zA-Z_][a-zA-Z_0-9]*
name  ::=     /[a-zA-Z_][a-zA-Z_0-9]*
```