

S-Lang Cheatsheet

Overview

- Interpreted programming language
- C-like syntax
- Dynamic typing
- Supports both procedural and object-oriented programming
- Comes with a standard library of useful functions

Variables

- Variables are dynamically typed
- Declaration not required
- Naming conventions:
 - Must start with a letter or underscore
 - Can contain letters, digits, and underscores
- Examples:

```
x = 5
str = "hello"
b = true
```

Functions

- Can take any number of arguments
- Can return any type
- Arguments are passed by value
- Examples:

```
function add(x, y) {
    return x + y
}
function print(str) {
    printf("%s\n", str)
}
```

Loops

- Supports while and for loops
- Examples:

```
while (x < 10) {
    x = x + 1
}
for (i = 0; i < 10; i++) {
    print(i)
}
```

Conditionals

- Supports if-else statements
- Examples:

```
if (x < 5) {  
    print("x is less than 5")  
} else {  
    print("x is greater than or equal to 5")  
}
```

File manipulation

- Supports reading and writing to files
- Examples:

```
f = fopen("file.txt", "r")  
line = fgets(f)  
fclose(f)
```

```
f = fopen("file.txt", "w")  
fprintf(f, "hello\n")  
fclose(f)
```

Resources

- [S-Lang Programming Language](#)
- [S-Lang Programming Language Wiki](#)
- [S-Lang Examples](#)