

Scheme Cheatsheet

This cheatsheet provides a brief overview of Scheme's unique features and syntax, including code blocks for variables, functions, loops, conditionals, file manipulation, and more.

Scheme is a dialect of Lisp and is known for its simple syntax, powerful macros, and functional programming paradigm. It is used in artificial intelligence research, education, and software development.

This cheatsheet is designed to serve as a quick reference guide for Scheme developers of all levels. It includes examples of common syntax and programming constructs, as well as a list of resources for further learning.

Unique Features

- Dialect of Lisp
- Simple syntax
- Powerful macros
- Functional programming paradigm
- Used in artificial intelligence research, education, and software development

Variables

```
; declaring a variable
(define variable_name value)

; dynamic typing
(define variable_name "string")
(define variable_name 123)
```

Functions

```
; defining a function
(define (function_name parameter1 parameter2)
  ; function body
  value)

; calling a function
(function_name argument1 argument2)
```

Loops

```
; do loop
(do () (condition)
  ; loop body)

; while loop
(while condition
  ; loop body)
```

```
; for loop
(for-each (lambda (variable_name)
  ; loop body)
  list_name)
```

Conditionals

```
; if statement
(if condition
  ; if body
  (if condition
    ; else if body
    ; ...
  )
  ; else body
)

; ternary operator
(if condition true_body false_body)
```

File Manipulation

```
; reading from a file
(call-with-input-file "filename"
  (lambda (file)
    (let loop ((line (read-line file)))
      (cond ((eof-object? line) #f)
            (else
             ; process line
             (loop (read-line file)))))))

; writing to a file
(call-with-output-file "filename"
  (lambda (file)
    (display "content" file)))
```

Resources

- [Official Scheme Website](#)
- [Scheme Documentation](#)
- [Scheme Style Guide](#)