Assembly Cheatsheet

Overview of unique features

- Low-level programming language
- Direct interaction with hardware
- No high-level abstractions
- Limited built-in functions
- Requires knowledge of CPU architecture

Variables

Assembly does not have variables in the same sense as high-level programming languages. Instead, it uses registers to store data.

```
; Move a value into a register
mov eax, 42

; Move a value from one register to another
mov ebx, eax
```

Functions

Assembly does not have functions in the same sense as high-level programming languages. Instead, it uses subroutines to perform specific tasks.

```
; Define a subroutine
mySubroutine:
  ; do something
  ret

; Call a subroutine
call mySubroutine
```

Loops

Assembly uses jump instructions to create loops.

```
; Define a loop
myLoop:
  ; do something
  jmp myLoop
```

Conditionals

Assembly uses conditional jump instructions to create if-else statements.

```
; Define an if-else statement
cmp eax, ebx
jg greaterThan
jmp lessThan

greaterThan:
  ; do something
  jmp endIf

lessThan:
  ; do something else
  jmp endIf

endIf:
```

Interacting with hardware

Assembly uses specific instructions to interact with hardware.

```
; Set up a digital pin
mov al, 0xFF
out 0x02, al

; Turn on and off an LED
mov al, 0x01
out 0x02, al
mov al, 0x00
out 0x02, al
```

Resources

- x86 Assembly documentation
- NASM tutorial
- Assembly language forum for community support and troubleshooting.