Smalltalk Cheatsheet

A concise reference for Smalltalk syntax, core concepts, and common idioms, designed to assist both beginners and experienced developers.

Core Syntax & Concepts

Basic Syntax

Message Sending (Method Call)	`receiver message`
	`receiver message: argument`
	`receiver message: arg1 message2: arg2`

Assignment	`variable := expression.`
Blocks (Anonymous Functions)	`[ expression. expression ]` `[:arg1 :arg2 | expression]`
Return	`^ expression`
Comments	`"This is a comment"`
Cascading	`object message1; message2; message3.`

Literals

Integers	`123`, `-456`
Floating-Point Numbers	`3.14`, `-0.001`
Characters	`$a`, `$Z`
Strings	`'Hello, world!'`
Symbols	`#mySymbol`, `#+`
Arrays	`#(1 2 3)`, `#( 'a' 'b' 'c' )`

Keywords

`true`, `false`, `nil`

Control Structures

Conditional Statements

If True	`condition ifTrue: [ expression ]`
If False	`condition ifFalse: [ expression ]`
If True: Else:	`condition ifTrue: [ trueExpression ] ifFalse: [ falseExpression ]`
Boolean Logic	`and:`, `or:`, `not` (as a message to a boolean receiver)

Loops

While True	`[ condition ] whileTrue: [ expression ]`
While False	`[ condition ] whileFalse: [ expression ]`
Times Repeat	`5 timesRepeat: [ expression ]`
Integer to:do:	`1 to: 10 do: [ :i | expression ]`
Collection do:	`#(1 2 3) do: [ :each | Transcript show: each printString ]`

Collections

Common Collection Operations

Adding elements	`collection add: element`
Removing elements	`collection remove: element`
Checking for element existence	`collection includes: element`
Getting the size	`collection size`
Testing if empty	`collection isEmpty`
Iterating	`collection do: [ :each | expression ]`

Specific Collection Types

Arrays	`Array new: 5. Array with: 1 with: 2 with: 3.`
Dictionaries	`Dictionary new. dictionary at: 'key' put: 'value'.`
Sets	`Set new. set add: 'element'.`
Ordered Collections	`OrderedCollection new. orderedCollection add: 'element'.`

Classes and Objects

Defining a Class

```
Object subclass: #MyClass
    instanceVariableNames: 'instanceVar1 instanceVar2'
    classVariableNames: ''
    poolDictionaries: ''
    category: 'MyCategory'
```

Instance variables are declared within the `instanceVariableNames:` field.

Methods

Instance Method Definition:

```
!MyClass methodsFor: 'accessing'!
myMethod
    ^ instanceVar1
! !
```

Class Method Definition:

```
!MyClass class methodsFor: 'instance creation'!
new
    ^ super new initialize
! !
```

Method categories help organize methods within a class definition.

Object Creation

`MyClass new`

Often, the `initialize` method is overridden to perform object setup.