Cheat Sheet #1

Hello World

Algorithm 1 Hello World

```
  public static void main()
  {
        System.out.println("Hello World!");
  }
```

Development

Editing

Using IDE for features like

- Code Completion
- Syntax checking
- Refactoring
- Automatic Code Generation

Common IDEs

- Eclipse
- Netbeans
- Intellij IDEA

Compiling

Using the compiler creates Bytecode out of sourcecode (.java → .out).
Use command line compiler *javac* or IDE.

Running

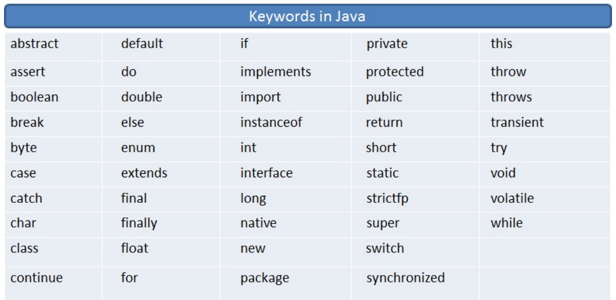
Bytecode can be run in the Java VM.
Use command line *java* or IDE.

Debugging

Jump into code at a certain point during runtime.
Set Breakpoints to specify debug entry points.
Observe current variable state.
Step through code line by line.

Reserved Names

There are round about 54 reserved names that cannot be used as identifiers in JAVA.



Keywords in Java				
abstract	default	if	private	this
assert	do	implements	protected	throw
boolean	double	import	public	throws
break	else	instanceof	return	transient
byte	enum	int	short	try
case	extends	interface	static	void
catch	final	long	strictfp	volatile
char	finally	native	super	while
class	float	new	switch	
continue	for	package	synchronized	

(http://www.f5java.com/images/java-tutorial-keywords-in-java.jpg)

Naming Conventions

Package names lower case.
Class names start capital, use CamelCase and describe the purpose of the class.
Function names start lower case, use CamelCase and describe its behavior.
Variable names start lower case and describe the data stored in the variable.
Interface names start capital, use CamelCase and describe an adjective the Interface provides.
Constant names have only capital letters and multiple words are separated by underscores.
Boolean **getters** have prefix "is", all other "get".
Setters have prefix "set".

Program Layout

Packages contain *classes* or other *packages*.
Classes contain *variables* and *functions*.
Every *class* contains a **constructor**.
Only *classes* containing a **main** method can be run.

Comments

Inline Comments → //
Multi Line Comments → /* * */
JavaDOC → /** * */
Try to avoid comments. If a code block has to be commented, **extract a method** with descriptive name instead. Use JavaDOC on API (public methods).

Coding conventions

There are no universal coding conventions. There are popular ones and rather specific ones. The important thing is to **agree on a set of rules** with your team!

- **Indentations** - use 2 spaces
- Open **parenthesis** in new line
- Use **Private Members** - use only getters and setters
- Do not use **Parameters as return values**
- Use **TDD** or at least **write tests** for each class afterwards
- Use **empty lines** for vertical grouping of variables
- Keep classes short → single responsibility
- Keep functions short → A function should do one thing only