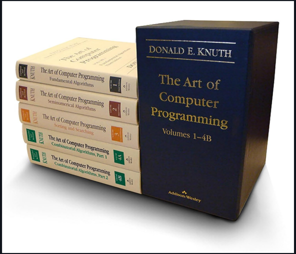
Linters & Formatters

Mitsiu Alejandro Carreño Sarabia


DONALD E. KNUTH
The Art of Computer Programming
Volumes 1–4B
Addison-Wesley
The Art of Computer Programming
Fundamental Algorithms
Seminumerical Algorithms
Sorting and Searching
Combinatorial Algorithms, Part 1
Combinatorial Algorithms, Part 2

Agenda

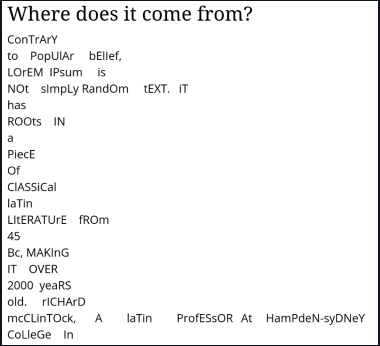
```
├── Formatters
├── Linters
├── Formatters & linters
├── Name conventions
├── Java implementation
└── Examples
```

Formatters

Coding style is how your code looks and it is an important part of writing code as a professional. Whether you're writing JavaScript or CSS or any other language, deciding how your code should look is an important part of overall code quality.

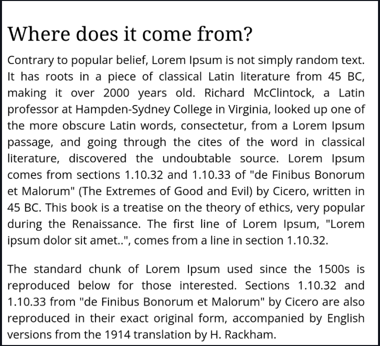
- Who we write code for?

Formatters



Where does it come from?

ConTrArY
to PopUlAr bEIlef,
LOrEM IPsum is
NOt sImpLy RandOm tEXT. iT
has
ROOts IN
a
PiecE
Of
ClASSiCal
laTin
LItERATUrE fROm
45
Bc, MAKInG
IT OVER
2000 yeaRS
old. rICHArD
mcCLinTOck, A laTin ProfESsOR At HamPdeN-syDNeY
CoLleGe In



Where does it come from?

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.

Formatters

Coding style is made up of numerous small decisions based on the language:

- How and when to use comments,
- Tabs or spaces for indentation (and how many spaces),
- Appropriate use of white space,
- Proper naming of variables and functions,
- Code grouping an organization,
- Patterns to be used,
- Patterns to be avoided.

Linters

Lint is the computer science term for a static code analysis tool used to flag programming errors, bugs, stylistic errors and suspicious constructs.

Formatter & linter

Formatting rules

- Maximum line length
- Disallow mixed spaces and tabs for indentation
- Enforce consistent spacing before and after keywords
- Enforce consistent comma style
- Etc

Code-quality rules

- Disallow unused variables
- Disallow declarations in the global scope
- Branching fall through
- Etc

In summary, formatters applies code style rules and linter detect logic errors.

Name conventions

Name conventions

- Variables => cammelCase

```
String nombre = "POO"
String nombreMateria =
"POO"
```

- Functions => cammelCase

```
static void printGreet(){
    ...
}
```

- Parameters => cammleCase

```
static int addTwo(int
firstNum){
    return firstNum + 2;
}
```

- Clases => PascalCase

```
class MyClass {
    ...
}
```

Java implementation

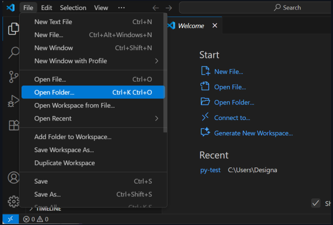
What is implementation?

Java implementation

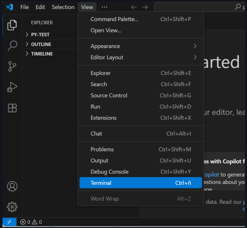
Terminal review A) VS Code

VS code

- Open the folder in vs code



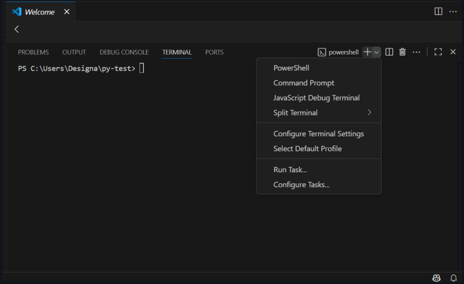
- Open a new terminal



Java implementation

Terminal review A) VS Code

- Choose between Powershell or cmd



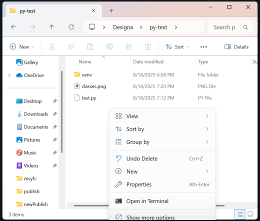
```
PS C:\Users\Designa\py-test>
```

Java implementation

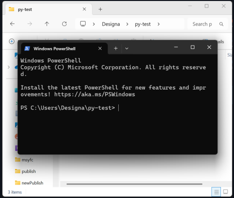
Terminal review B) File explorer PS

File explorer and Powershell

- Secondary click "Abrir en terminal"



Name	Date modified	Type
venv	8/18/2025 6:58 PM	File folder
classes.png	8/18/2025 7:20 PM	PNG File
test.py	8/18/2025 7:12 PM	PY File



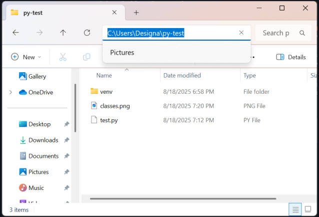
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserve
d.

Install the latest PowerShell for new features and impr
ovements! https://aka.ms/PSWindows

PS C:\Users\Designa\py-test>
```

Java implementation

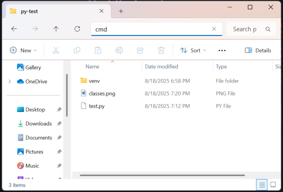
Terminal review C) File explorer cmd

File explorer and cmd

- Click path bar



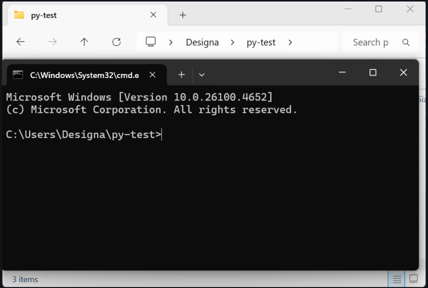
- Erase current path and write "cmd"



Java implementation

Terminal review C) File explorer cmd

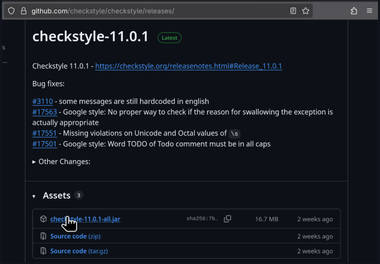
File explorer and cmd



```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Designa\py-test>
```

Java implementation

1. You need to know the file path to your <file>.java
2. You need to download checkstyle-11.0.1-all.jar (https://github.com/checkstyle/checkstyle/releases/)


github.com/checkstyle/checkstyle/releases/

checkstyle-11.0.1 Latest

Checkstyle 11.0.1 - https://checkstyle.org/releasenotes.html#Release_11.0.1

Bug fixes:

#3110 - some messages are still hardcoded in english
#17563 - Google style: No proper way to check if the reason for swallowing the exception is actually appropriate
#17551 - Missing violations on Unicode and Octal values of `\s`
#17501 - Google style: Word TODO of Todo comment must be in all caps

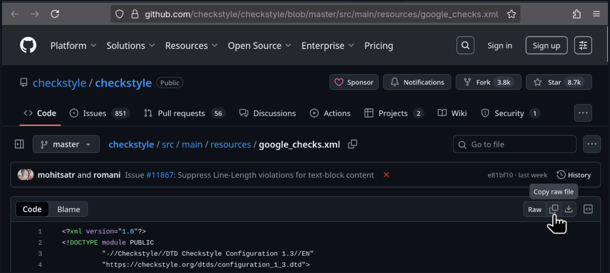
▸ Other Changes:

▾ Assets 3

checkstyle-11.0.1-all.jar	sha256:7b...	16.7 MB	2 weeks ago
Source code (zip)			2 weeks ago
Source code (tar.gz)			2 weeks ago

Java implementation

3. Download google_checks.xml (https://github.com/checkstyle/checkstyle/blob/master/src/main/resources/google_checks.xml)



```
<?xml version="1.0"?>
<!DOCTYPE module PUBLIC
          "-//Checkstyle//DTD Checkstyle Configuration 1.3//EN"
          "https://checkstyle.org/dtds/configuration_1_3.dtd">
```

Java implementation

4. All files should be in the same directory:
    - <file>.java
    - checkstyle-11.0.1-all.jar
    - google_checks.xml

Java implementation

Terminals ofter feature autocompletion via tab

```
java -jar checkstyle-11.0.1-all.jar -c
google_checks.xml <file>.java
```

Examples

```java
class e9Checkstyle {
    ...
}
```

```
E9Checkstyle.java:1:7: Type name 'e9Checkstyle' must match pattern
'^[A-Z][a-zA-Z0-9]*$'. [TypeName]
```

```java
class E9Checkstyle {
    ...
}
```

Examples

```java
// File: e9Checkstyle.java
class E9Checkstyle {
    ...
}
```

e9Checkstyle.java:1:1: The name of the outer type and the file do not match. [OuterTypeFilename]

```java
// File: E9Checkstyle.java
class E9Checkstyle {
    ...
}
```

Examples

```
class E9Checkstyle {
public static void main(String[] args) {
    ...
}
}
```

```
E9Checkstyle.java:2:1: 'method def modifier' has incorrect indentation
level 0, expected level should be 2. [Indentation]
E9Checkstyle.java:4:1: 'method def rcurly' has incorrect indentation
level 0, expected level should be 2. [Indentation]
```

```
class E9Checkstyle {
    public static void main(String[] args) {
        ...
    }
}
```

Examples

```java
class E9Checkstyle {
    public static void main(String[] args){
        ...
    }
}
```

```
E9Checkstyle.java:2:41: WhitespaceAround: '{' is not preceded with
whitespace. [WhitespaceAround]
```

```java
class E9Checkstyle {
    public static void main(String[] args) {
        ...
    }
}
```

Examples

```
class E9Checkstyle {
    public static void main(String[] Args) {
        ...
    }
}
```

E9Checkstyle.java:2:36: Parameter name 'Args' must match pattern '^**a-z** ([a-z0-9][a-zA-Z0-9]*)?$'. [ParameterName]

```
class E9Checkstyle {
    public static void main(String[] args) {
        ...
    }
}
```

Resources

https://google.github.io/styleguide/javaguide.html
https://github.com/checkstyle/checkstyle/releases/
https://github.com/checkstyle/checkstyle/blob/master/src/main/resources/google_checks.xml

References

https://web.archive.org/web/20230606032819/http://www.sublimelinter.com/en/v3.10.10/about.html https://prettier.io/docs/comparison
https://www.smashingmagazine.com/2012/10/why-coding-style-matters/