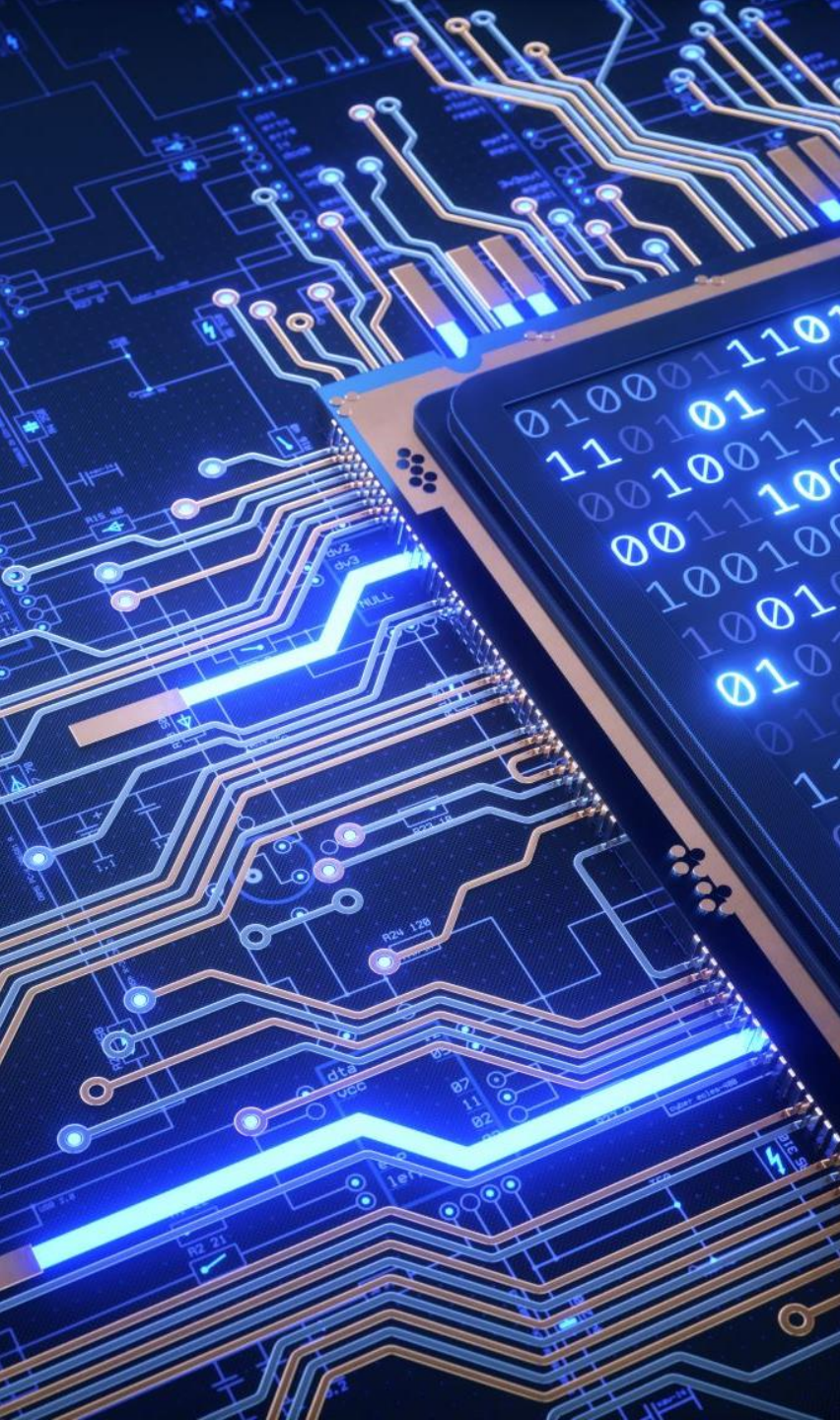


# Podstawowe narzędzia developerskie testera

Wykład 4 – Powtórzenie i Podsumowanie



# O czym mówiliśmy

Systemie Operacyjnym

Konsola

Zarządzaniu wersją

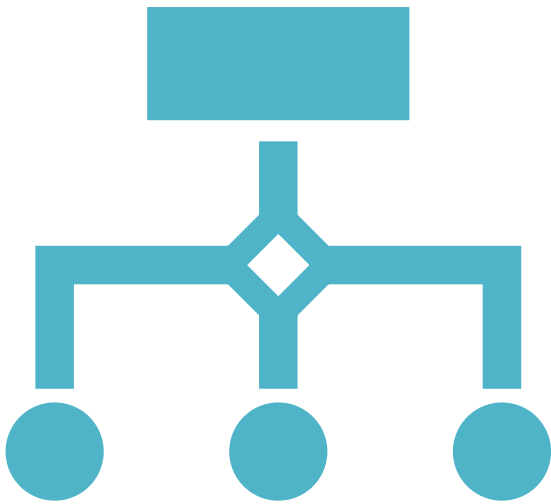
Runtime

IDE

Środowisko lokalne i zdalne

# System Operacyjny

„System operacyjny (ang. operating system, skrót OS) – oprogramowanie zarządzające systemem komputerowym, tworzące środowisko do uruchamiania i kontroli zadań.”



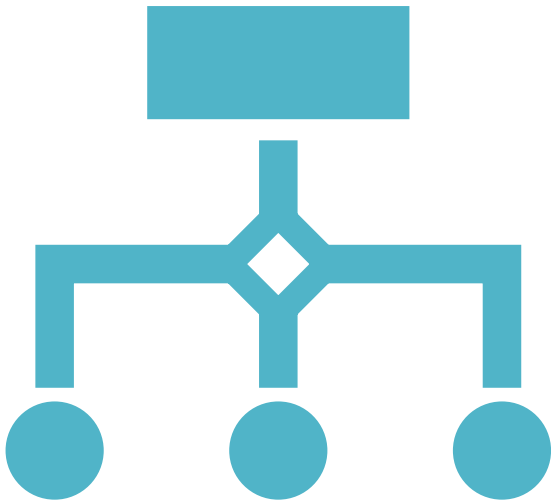
# System Operacyjny

Co oznacza to dla testera

Jest to zmienna wpływająca na testy

Jest to środowisko wykonywania testów

Jest to narzędzie wykonywania testów



## Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is installed.

- ☒ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ☒ .NET Framework 4.8 Advanced Services
- ☐ Active Directory Lightweight Directory Services
- ☒ Containers
- ☐ Data Center Bridging
- ☐ Device Lockdown
- ☐ Guarded Host
- ☒ Hyper-V
- ☒ Internet Explorer 11
- ☐ Internet Information Services
- ☐ Internet Information Services Hostable Web Core
- ☐ Legacy Components
- ☒ Media Features
- ☐ Microsoft Defender Application Guard
- ☐ Microsoft Message Queue (MSMQ) Server
- ☒ Microsoft Print to PDF
- ☒ Microsoft XPS Document Writer
- ☐ MultiPoint Connector
- ☒ Print and Document Services
- ☒ Remote Differential Compression API Support
- ☐ Services for NFS
- ☐ Simple TCP/IP services (i.e. echo, daytime etc)
- ☐ SMB 1.0/CIFS File Sharing Support
- ☐ SMB Direct
- ☐ Telnet Client
- ☐ TFTP Client
- ☒ Virtual Machine Platform
- ☐ Windows Hypervisor Platform

# System Operacyjny

## Windows Features

Environment Variables

User variables for Kercon

Variable	Value
ChocolateyLastPathUpdate	132585852335505899
OneDrive	C:\Users\Kercon\OneDrive
OneDriveConsumer	C:\Users\Kercon\OneDrive
Path	C:\Users\Kercon\AppData\Local\Programs\Python\Python35-32\Scripts\;C:\Users\Kerc...
TEMP	C:\Users\Kercon\AppData\Local\Temp
TMP	C:\Users\Kercon\AppData\Local\Temp

New... Edit... Delete

System variables

Variable	Value
_PSLockDownPolicy	0
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
FP_NO_HOST_CHECK	NO
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Python39\Scripts\;C:\Python39\;C:\Program Files (x86)\Razer Chroma SDK\bin;C:...

New... Edit... Delete

OK Cancel

# System Operacyjny

Windows Features

Environment Variables



```
C: > Windows > System32 > drivers > etc > hosts
1  # Copyright (c) 1993-2009 Microsoft Corp.
2  #
3  # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4  #
5  # This file contains the mappings of IP addresses to host names. Each
6  # entry should be kept on an individual line. The IP address should
7  # be placed in the first column followed by the corresponding host name.
8  # The IP address and the host name should be separated by at least one
9  # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #      102.54.94.97      rhino.acme.com      # source server
17 #      38.25.63.10      x.acme.com          # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1          localhost
21 #   ::1                localhost
22 # Added by Docker Desktop
23 192.168.1.178 host.docker.internal
24 192.168.1.178 gateway.docker.internal
25 # To allow the same kube context to work on the host and the container:
26 127.0.0.1 kubernetes.docker.internal
27 # End of section
28 |
```

# System Operacyjny

Windows Features

Environment Variables

Plik Host

# Konsola

Mówiliśmy o:

Wierszu Poleceń

PowerShell



# Top 5 poleceń

CD



Set-Location

DIR



Get-ChildItem

Ipconfig



Get-NetIPAddress

Ping



Test-Connection

tasklist



Get-Process

taskkill



Stop-Process

# Przykłady

CD:

```
cd C:\Users\
```

```
cd .\Student\
```

ipconfig

```
ipconfig /flushdns
```

```
ipconfig /renew
```

# Przykłady

Ping

```
Ping www.wp.pl
```

```
ping /t www.wp.pl
```

tasklist

```
tasklist /fi "IMAGENAME eq chrome.exe,,
```

```
tasklist | find "chrome,,
```

taskkill

```
taskkill /f /fi "IMAGENAME eq chrome.exe"
```

# Zarządzanie wersją

Powiedzieliśmy o:

- Co to

- Czemu używamy

- Czym jest GIT

- Podstawy pracy z GIT

# Przypomnienie Zarządzanie Wersją

System Kontroli Wersji z języka angielskiego Version Control System (VCS) jest oprogramowaniem śledzącym zmiany w plikach oraz umożliwia ich przewrócenie.



# Przypomnienie Zarządzanie Wersją

## Lokalne

My przechowujemy zarazem pliki jak ich różnice/kopie

## Zdalne

My posiadamy dostęp do najnowszej wersji plików nad którymi obecnie pracujemy ale zapisujemy zmiany na zdalnym serwerze

## Rozproszone

Każdy członek posiada kopie nie tylko najnowszej wersji plików ale całą historię.

# Przypomnienie Zarządzanie Wersją

Większość nowoczesnych firm korzysta zdalnych repozytoriów czy to scentralizowanych czy to z rozproszonych.

Wykorzystujemy System Kontroli wersji przykładowo do:

- Bezpiecznej pracy z kodem bez narażania się na nieodwracalne zmiany

- Równoległej pracy wielu osób w sposób niekolizyjny

- Rozróżniania wersji Produkcyjnej od wersji Rozwojowej oprogramowania

# Przypomnienie Zarządzanie Wersją

## Git

Jest to Narzędzie kontroli wersji typu rozproszonego

Pozwala na pracę w systemie gałęzi tzw. Nieliniowy

Wszystkie operacje na repozytorium są lokalne

Zmienione pliki trzeba zadeklarować do śledzenia (Stage)

Zadeklarowane pliki można powierzyć/zatwierdzić (Commit)

# GIT

Powiedzieliśmy:

- Czym jest commit

- Jak dołączamy zmiany do commit

- Jak wykorzystać checkout do cofnięcia się do commit

- Czym jest branch

- Jak przełączanie między branchami działa

- Czym jest merge

- Czym są konflikty

# GIT

Powiedzieliśmy:

Czym jest commit

*Jest to migawka czyli stan wszystkich wyznaczonych plików w momencie jej tworzenia*

*Musi być podpisany*

Podpis jest brany z konfiguracji:

`git config --global user.name`

`git config --global user.email`

*Posiada unikalny identyfikator w postaci hash sha1*

*Posiada wiadomość*



# GIT

Powiedzieliśmy:

Jak dołączamy zmiany do commit

*Poleceniem*

`git add <nazwa pliku>`

`git add .`

*Przez interfejs graficzny*

# GIT

Powiedzieliśmy:

Jak wykorzystać checkout do cofnięcia się do commit

*git checkout <hash commita>*

*Headless state*

# GIT

Powiedzieliśmy:

Czym jest branch

*Gałąź to pozwala na niezależne równoległe rozwijanie kodu który z powrotem trafi do głównej gałęzi*

*Zależności od firmy gałęzie mogą być używane do przechowywania twardej wersji aplikacji*

*Ryzyka branchowania*

Przenoszenie plików między branchami

Dziki rozdzielenia

Jak stworzyć branch

*Polecenie `git branch <newbranchname>`*

*Polecenie `git branch <newbranchname> <oldbranchname>`*

*Przy użyciu GUI*

# GIT

Powiedzieliśmy:

Jak przełączanie między branchami działa

*Powiedzieliśmy o poleceniu*

`git switch <branchname>`

`git checkout <branchname>`

# GIT

Powiedzieliśmy:

Czym jest merge

*Nanoszenie zmian z jednego branch na drógi*

*Czym jest fast forward*

*Czym jest merge commit*

Czym są konflikty

*Skąd powstają*

*Binarne pliki vs Tekstowe*

*Jak git oznacza konflikt wewnątrz pliku*

*Jak rozwiązujemy konflikty*



```

PB-EDU-Tester1 > .gitignore
You, 3 weeks ago | 1 author (You)
1 # Logs
2 logs
3 *.log
4 npm-debug.log*
5 yarn-debug.log*
6 yarn-error.log*
7 lerna-debug.log*
8
9 # Diagnostic reports (https://nodejs.org/api/report.html)
10 report.[0-9]*.[0-9]*.[0-9]*.[0-9]*.json
11 | You, 3 weeks ago * Initial commit
12 # Runtime data
13 pids
14 *.pid
15 *.seed
16 *.pid.lock
17
18 # Directory for instrumented libs generated by jscoverage/JSCover
19 lib-cov
20
21 # Coverage directory used by tools like istanbul
22 coverage
23 *.lcov
24
25 # nyc test coverage
26 .nyc_output
27
28 # Grunt intermediate storage (https://gruntjs.com/creating-plugins#storing-task-files)
29 .grunt
30
31 # Bower dependency directory (https://bower.io/)
32 bower_components
33
34 # node-waf configuration
35 .lock-wscript
36
37 # Compiled binary addons (https://nodejs.org/api/addons.html)
38 build/Release
39
40 # Dependency directories
41 node_modules/
42 jspm_packages/
43

```

# GIT

Nie powiedzieliśmy czym jest:  
.gitignore

Zatem czym jest?

Automatyczne wykluczenie

*Plików*

Specyficznych

Jak i wzorcem

*Folderów*

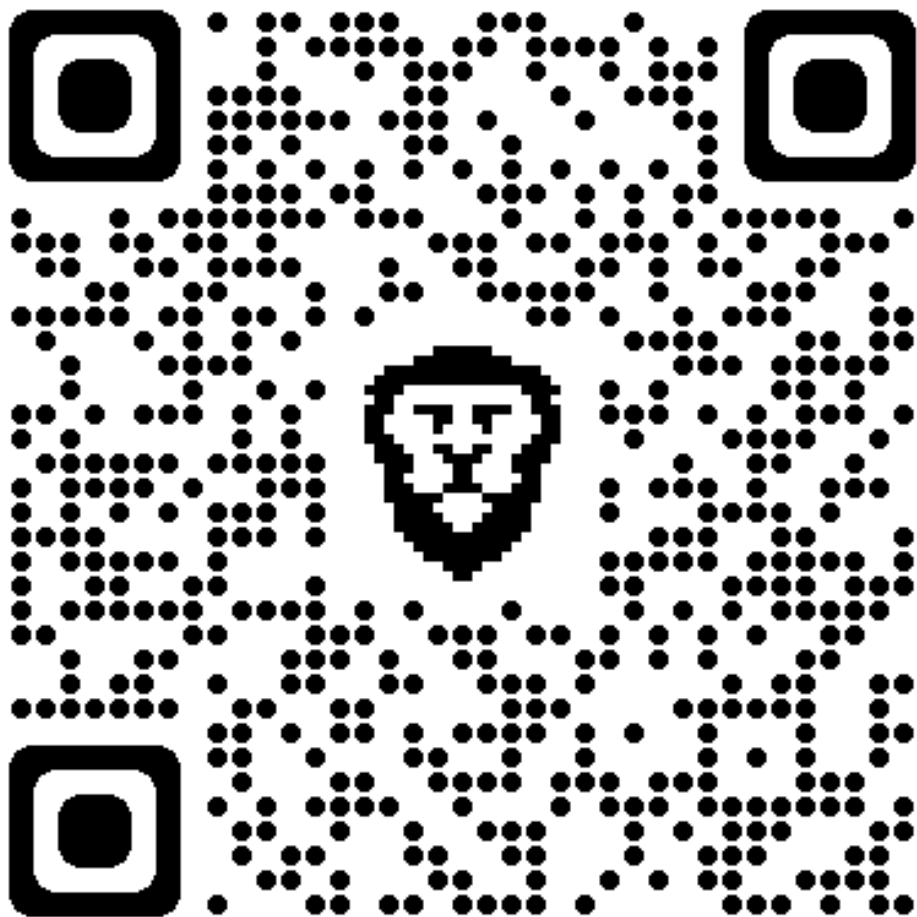


# RUNtime

Dostarcza zestaw  
oprogramowania/bibliotek  
(ekosystem) pozwalający na  
wykonywanie kodu

Powiedzieliśmy o wpływie  
runtime na testy

Przedstawiliśmy sobie node.js



IDE

VISUAL STUDIO CODE

# IDE

„Zintegrowane środowisko programistyczne, IDE (od ang. integrated development environment) – program lub zespół programów (środowisko) służących do tworzenia, modyfikowania, testowania i konserwacji oprogramowania.”

# IDE

Mieliśmy przyjemność

Użycia visual studio code jako:

*Edytora*

*Terminalu*

*Klienta graficznego GIT*

Zainstalowaliśmy dodatki:

*GitLens*

*Docker*



# Środowiska testowe

Podzieliliśmy je na

Lokalne

Zdalne

Oraz

Trwałe

Tymczasowe

# Przygotowanie środowiska

Powiedzieliśmy sobie o narzędziach która mogą przyspieszyć przygotowanie środowiska

Z poziomu developerskiego

*Dystrybutory paczek*

NPM

*GIT*

Z poziomu oprogramowania

*Chocolatey*

*Docker*