Developing inside a Container
Not this type of container!
I mean, this type of container!
= Modern (and portable) Development Environment
How it works

Source: https://code.visualstudio.com/docs/remote/containers
What if I need a specific version of go?
Demo

```json
1  {
2    "name": "Go Demo",
3    "image": "mcr.microsoft.com/vscode/devcontainers/go:1.16-bullseye"
4  }
```
Use case #1: Making bitmap fonts
Adafruit GFX Graphics Library

PaintYourDragon Merge pull request #268 from BlueAndj/master ...

..

- Makefile
  minor changes to build on Fedora
  6 years ago
- bdf2adafruit.py
  Trailing spaces removed.
  2 years ago
- fontconvert.c
  fontconvert missing header (#297)
  2 years ago
- fontconvert_win.md
  Update fontconvert_win.md
  6 years ago
- makefonts.sh
  Fonts FONTS fonts/1.1 release
  7 years ago
Use case #1: Making bitmap fonts

```json
 1  {
 2           "name": "GfxFontConverter",
 3           "build": {
 4               "dockerfile": "Dockerfile"
 5           },
 6           "settings": {},
 7           "extensions": [
 8               "ms-vscde.cpptools"
 9           ]
10    }
```
Use case #1: Making bitmap fonts

```
# X86  : debian-11, debian-10,
#       ubuntu-21.04, ubuntu-20.04, ubuntu-18.04
# ARM64 : debian-11, ubuntu-21.04, ubuntu-18.04

ARG VARIANT=ubuntu-21.04
FROM mcr.microsoft.com/vscode/devcontainers/cpp:0-{$VARIANT}

RUN apt-get update && \
    export DEBIAN_FRONTEND=noninteractive && \
    apt-get -y install --no-install-recommends \
    libfreetype-dev \
    wget
```
Use case #1 : Making bitmap fonts

✓ No need to install libraries on my Mac
✓ Easily available for others
Use case #2 : Building Websites
Use case #2: Building Websites

MkDocs
Project documentation with Markdown.

pymdownx.arithmatex
pymdownx.superfences
...
awesome-pages
macros
i18n
...
Use case #2: Building Websites

Files and folders structure:
Use case #2: Building Websites

devcontainer.json

```json
{
  "name": "MkDocs-Edu",
  "build": {
    "dockerfile": "Dockerfile"
  },
  "runArgs": [],
  "settings": {},
  "extensions": []
}
```
Use case #2: Building Websites

Dockerfile

```
1 ARG VARIANT=bullseye
2 FROM mcr.microsoft.com/vscode/devcontainers/python:0-$\{VARIANT\}
3
4 RUN apt-get update && \
5   export DEBIAN_FRONTEND=noninteractive && \
6   apt-get -y install --no-install-recommends \
7   imagemagick \
8   zip
9
10 WORKDIR /mkdocs-edu
11 COPY mkdocs-edu/* /mkdocs-edu/
12
13 RUN pip install poetry && \
14   poetry config virtualenvs.create false && \
15   poetry install
16
17 COPY scripts/serve.sh /usr/local/bin/serve
18 COPY scripts/build.sh /usr/local/bin/build
```
Use case #2: Building Websites

```toml
[tool.poetry]
name = "mkdocs-edu"
version = "0.1.0"
description = "Docker image for mkdocs-edu"
authors = ["Jacques Supcik <jacques.supcik@hefr.ch>"
license = "Apache-2.0"

[tool.poetry.dependencies]
python = "^3.9"
mkdocs = "^1.2.3"
mkdocs-material = "^8"
mkdocs-macros-plugin = "^0.6.3"
python-markdown-math = "^0.8"
pymdown-include = "^0.1.0"
mkdocs-awesome-pages-plugin = "^2.6.1"
jinja2-cli = "^0.8.2"

[build-system]
requires = ["poetry-core>=1.0.0"
build-backend = "poetry.core.masonry.api"
```
Use case #2: Building Websites

Dockerfile (GitHub Actions)

```bash
FROM python:bullseye

RUN apt-get update && \
    export DEBIAN_FRONTEND=noninteractive && \
    apt-get -y install --no-install-recommends \ 
    zip

RUN pip install poetry

COPY entrypoint.sh /entrypoint.sh

ENTRYPOINT ["/entrypoint.sh"]
```
Use case #2 : Building Websites

```python
1  site_dir="${1:-public}"
2  config="${2:-config/mkdocs.yml}"
3  week="${3:-999}"
4  solutions="${4:-999}"
5
6  cp -a .devcontainer/mkdocs-edu /
7  (cd /mkdocs-edu && poetry config virtualenvs.create false && poetry install)
8
9  python --version
10  mkdocs --version
11
12  echo "Filtering pages"
13  for i in $(find docs -name pages.j2); do
14      echo "Processing $i"
15      jinja2 -D week=$week -o $(dirname $i)/.pages $i
16  done
17
18  echo "Building site"
19  export SHOW_SOLUTION=${solutions}
20  mkdocs build --clean --config-file ${config} --site-dir ${site_dir}
```
Use case #2: Building Websites

https://ado.isc.heia-fr.ch (gitlab @HEIA)
https://se1.isc.heia-fr.ch (netlify)
https://mse-csel.github.io/website (github)
Use case #2 : Building Websites

✓ Reproducible build
✓ Works for Windows/MacOS/Linux
✓ Similar configuration for GitHub Actions
Use case #3 : MSE Lecture - CSEL1
Use case #3: MSE Lecture - CSEL1

```yaml
services:
  toolchain:
    build:
      context: toolchain
      dockerfile: Dockerfile
    volumes:
      - ..:/workspace:cached
      - buildroot:/buildroot
      - rootfs:/rootfs

volumes:
  buildroot: {}
  rootfs: {}
Use case #3: MSE Lecture - CSEL1

toolchain/Dockerfile

```bash
1 ARG VARIANT=ubuntu-21.04
2 FROM mcr.microsoft.com/vscode/devcontainers/cpp:0-#${VARIANT}
3
4 RUN apt-get update && \
5     export DEBIAN_FRONTEND=noninteractive && \
6     apt-get -y install --no-install-recommends \
7     autoconf \
8     automake \
9     ... \
10    u-boot-tools \
11    wget
12
13 COPY scripts/* /usr/local/bin/
14 RUN chmod +x /usr/local/bin/*
```
Use case #3: MSE Lecture - CSEL1

```json
{
  "name": "cse1",
  "dockerComposeFile": "docker-compose.yml",
  "service": "toolchain",
  "workspaceFolder": "/workspace",
  "shutdownAction": "stopCompose",
  "settings": {},
  "extensions": [
    "ms-vscode.cpptools"
  ]
}
```
Use case #3 : MSE Lecture - CSEL1

✓ VMware no longer needed
✓ Works on Windows/MacOS/Linux
✓ The system is fast and responsive
✓ Resources are efficiently used
✓ Nice developer experience : Opening VS Code automatically starts all containers
✓ System easily adapted for other lectures
Some restrictions
Caution: Code is not always portable

```python
def is_executable(full_path: str) -> bool:
    """
    os.X_OK will always return true on windows. Use git to check file mode.
    :param full_path: file full path
    :return: True is it's an executable file
    """
    if sys.platform == 'win32':
        return _check_git_filemode(full_path)
    return os.access(full_path, os.X_OK)
```

os.X_OK also always returns `true` if the file is a mounted volume in Docker!

https://github.com/espressif/esp-idf/blob/a82e6e63d98bb051d4c59cb3d440c537ab9f74b0/tools/ci/idf_ci_utils.py#L74
The Future is in the Cloud

Open Source

SaaS or Self-Hosted

Secure by design
Base images

vscode/devcontainers/base  vscode/devcontainers/ruby
vscode/devcontainers/javascript-node  vscode/devcontainers/go
vscode/devcontainers/typescript-node  vscode/devcontainers/java
vscode/devcontainers/universal  vscode/devcontainers/dotnetcore
vscode/devcontainers/python  vscode/devcontainers/dotnet
vscode/devcontainers/cpp  vscode/devcontainers/anaconda
vscode/devcontainers/php  vscode/devcontainers/miniconda
vscode/devcontainers/rust  vscode/devcontainers/jekyll

Microsoft Container Registry (MCR) : https://github.com/microsoft/containerregistry
Tag List (e.g. vscode/devcontainers/go)

```json
{
    "name": "vscode/devcontainers/go",
    "tags": [
        "0-1-bullseye",
        "0-1.18",
        "0-1.18-bullseye",
        "0-1.18-buster",
        "0-bullseye",
        "0-buster",
        ...,
        "1",
        "1-bullseye",
        "1-buster",
        "1.18",
        "1.18-bullseye",
        "bullseye",
        "latest"
    ]
}
```

https://mcr.microsoft.com/v2/vscode/devcontainers/go/tags/list
Repository of Dockerfiles

https://github.com/microsoft/vscode-dev-containers
Links

Link to this presentation

MSE Lecture - CSEL1
https://github.com/mse-csel/csel-workspace

Making bitmap fonts
https://github.com/heia-fr/gfx-fontconvert

Building Websites
https://github.com/mse-csel/website

Devcontainer Cheat Sheet
https://cheatography.com/supcik/cheat-sheets/devcontainer/