Podstawowe technologie komunikacji technicznej

Część 4, 13.02.2021

Marta Bartnicka, Daniel Barrio Fierro

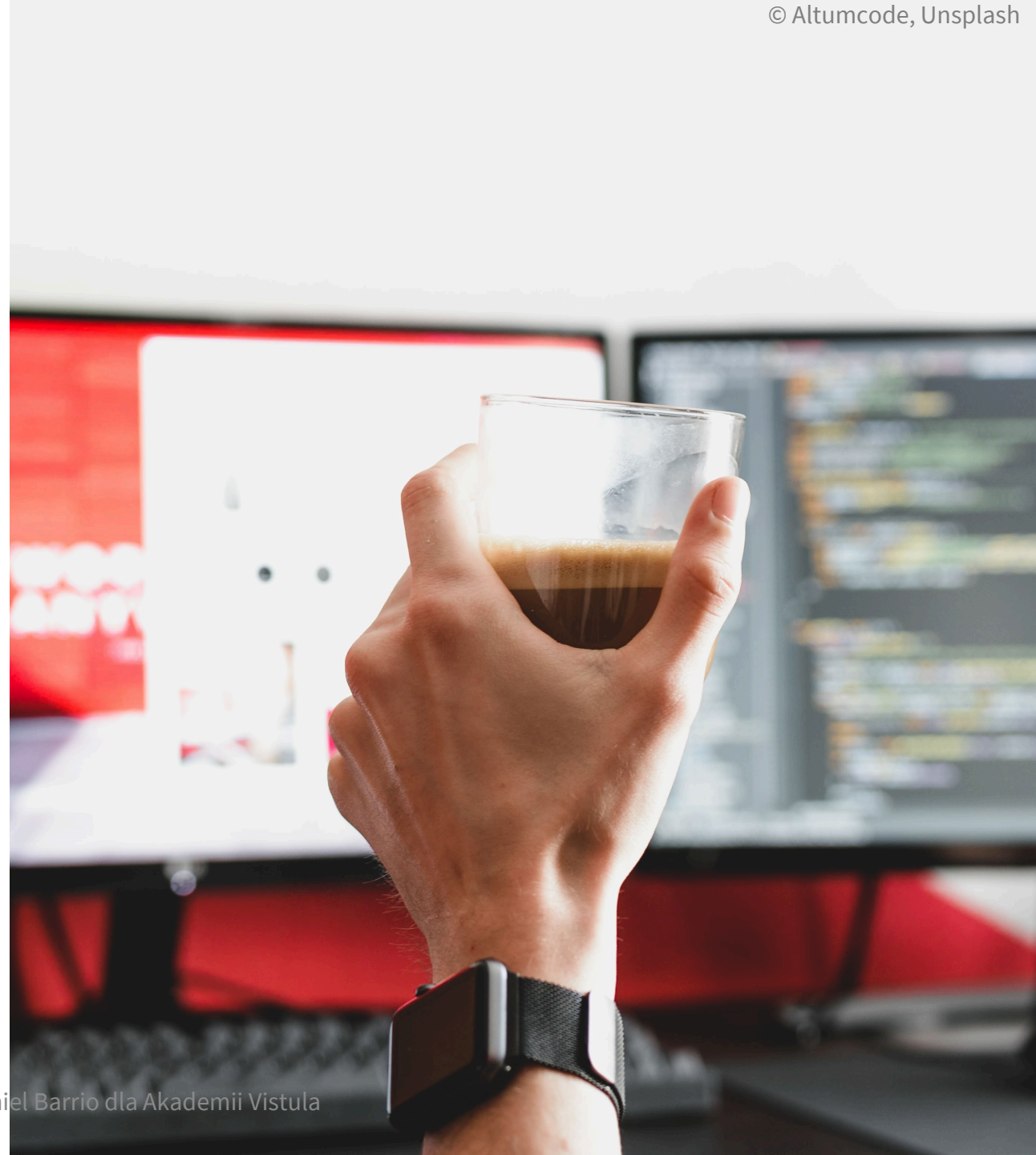
Spis treści

Przerwy

- 10:00 – 10:10
- 11:00 – 11:10
- 12:00 – 12:10

- Przerwa obiadowa:
  13:00 – 13:45

Koniec: 15:00



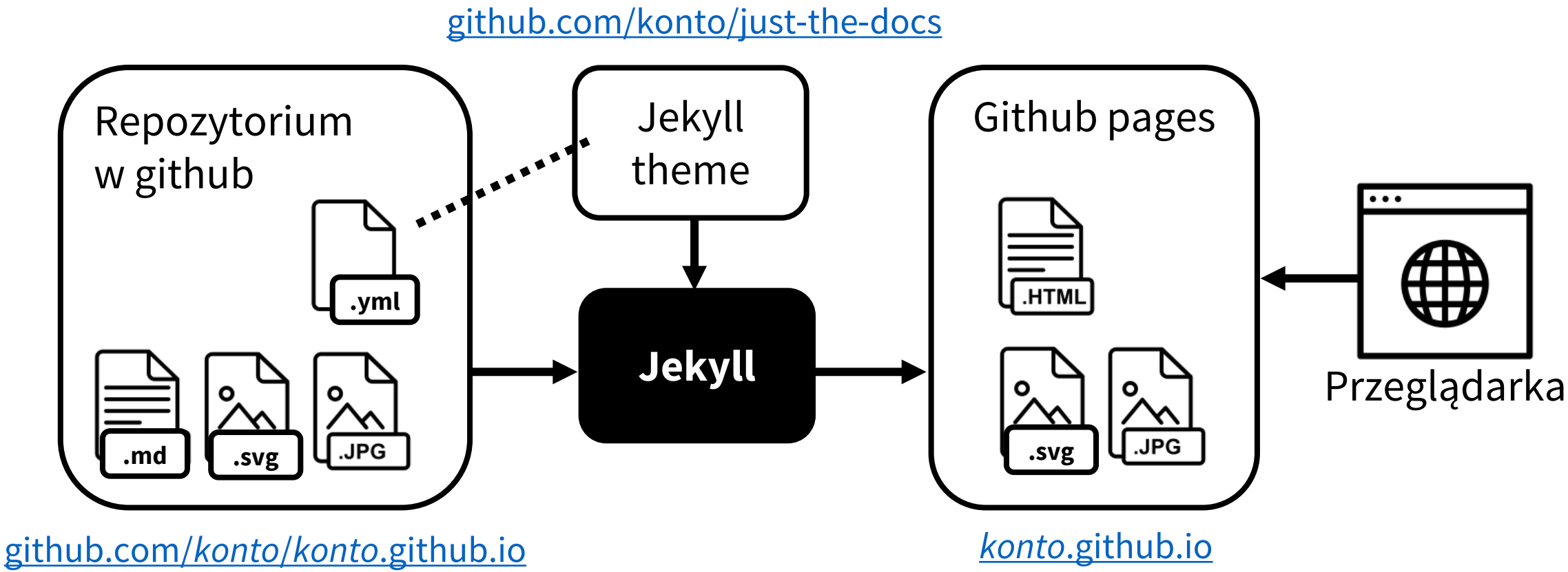
Co będzie potrzebne na dzień 4:

- Visual Studio Code (dzień 1)
- GitHub Desktop Client (dzień 2)
- konto.github.io (dzień 3)
- kawałek dokumentacji do VS Code, Markdowna i GitHuba (tyle, ile dotąd napisaliście)
- opcjonalnie: fork just-the-docs (dzień 3)

Publikowanie dokumentacji online

- Publikacja (przez GitHub pages)
- Struktura (just-the-docs)
- Ustawienia publikacji (YML)
- Ustawienia wizualne (CSS)

Jak to działa



Struktura *konto/konto*.github.io

- _includes : folder dla SVGs
  - image.svg
- assets
  - Images : folder dla JPGs, PNGs
    - image.jpg
- docs
  - chapter
    - file.md
    - index.md : ojciec dla chapter/file.md
  - plik.md

index.md

_config.yml : plik konfiguracyjny dla Jekyll

Przykładowe pliki I

_config.yml

```
name: dowolny
title: tytuł strony (widoczny na stronie)
description: opis strony
remote_theme: konto/just-the-docs (konto = nazwa naszego konta!)

color_scheme: light (może być też “dark”, albo jakiś inny stworzony pod konto/just-the-docs/_sass/color_schemes)

aux_links:
  "Source repository on GitHub":
    - "//github.com/konto/konto.github.io“ (konto = nazwa naszego konta!)

back_to_top: true
back_to_top_text: "Back to top“

footer_content: "This documentation is a student project for the 2020-2021
Technical Communication course at Vistula University (Warsaw)."
```

Przykładowe pliki II

index.md

```
---
layout: default (Nie zmieniać! Bardzo ważne dla Jekyll)
title: Introduction (Widoczne w Table of Contents)
nav_order: 1 (index.md to pierwsza strona w naszym Table of Contents)
description: “Description of this chapter"
permalink: /
---


# Introduction (Widoczne w treści)

Markdown text goes here.
```

Przykładowe pliki III

docs/plik.md

```
---
layout: default
title: Title
nav_order: 2
---


# Title

Markdown text goes here.
```

Przykładowe pliki IV

chapter/index.md

```
---
layout: default
title: Chapter title (zapamiętaj ten tytuł podczas konfigurowania stron potomnych)
nav_order: 3 (pozycja rozdziału w głównym Table of Contents)
has_children: true
---

# Title

Markdown text goes here.
```

Przykładowe pliki V

chapter/plik.md

```
---
layout: default
title: Title
parent: Chapter title (tytuł pliku chapter/index.md)
nav_order: 1 (pozycja tej strony pod jej elementem nadrzędnym “Chapter title”)
---

# Title

Markdown text goes here.
```

Jak zmienić wygląd?

- Kopiujemy plik z [komunikacjatechnicznavistula/just-the-docs/_sass/color_schemes/vistula.scss]
- Wklejamy plik do *konto/just-the-docs/_sass/color_schemes*
- Edytujemy plik (na przykład zmieniając kolory)
- W *konto/konto.github.io/_config.yml* dodajemy:

```
color_scheme: nazwa_pliku
```

Ściąga do just-the-docs (1)

- Jak konfigurujemy pliki źródłowe? https://pmarsceill.github.io/just-the-docs/docs/configuration/
- Jak konfigurujemy nawigację? https://pmarsceill.github.io/just-the-docs/docs/navigation-structure/
- Jak zmieniamy wygląd szablonu? https://pmarsceill.github.io/just-the-docs/docs/customization/

Nanosimy zmiany na naszej kopii szablonu just-the-docs! (fork)

Ściąga do just-the-docs (2)

- Jak dodajemy SVG jako grafikę wektorową w pliku markdown?

  ```
  {% include plik.svg %}
  ```
  (uwaga: tylko w Jekyll/Kramdown)

- Jak podkreślamy tekst?

  ```
  *underlined text*{: style="text-
  decoration: underline; font-style: normal;"}
  ```

- Jak skalujemy obrazek?

  ```
  ![img](/assets/images/image.jpg){: style="width: 200px;"}
  ```

- Jak dodajemy plik PDF?

  ```
  [pdf](/assets/pdfs/markdown-cheatsheet.pdf)
  ```

Jak dodać drugie repozytorium do strony HT Magdalena Niedźwiecka-Pruszkowska

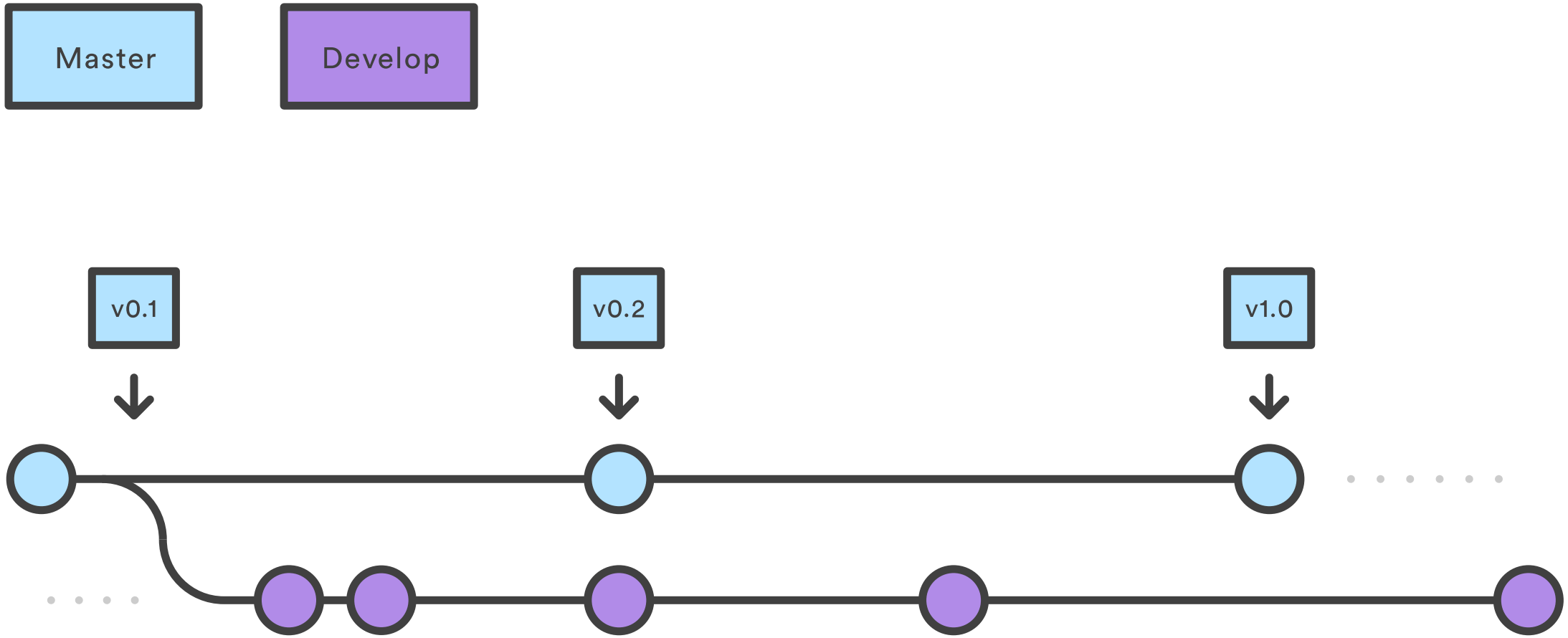
1. Wchodzę do repozytorium publicznego, z którego chcę puścić stronę
2. Klikam na Settings
3. Zjeżdżam na sam dół do GitHub pages
4. Zmieniam ustawienia: z none na master lub main lub inny branch
5. Save (po prawej stronie od nazwy brancha)
6. Wracam na stronę główną repozytorium (Code < >)
7. Po prawej stronie jest sekcja Environments - GitHub pages są Active
8. KLIK (na Environments)
9. Kliknięcie przenosi na stronę Deployments/Activity log
10. Klikam na View deployment
11. TADAM!

Redagowanie dokumentacji online

- Branch
- Sprawdzanie wg STE
- Merge
- Publikacja po zmianach
- *Pracujemy na repozytoriach GitHub Pages (konto.github.io)*
- *Grupy 2-osobowe*

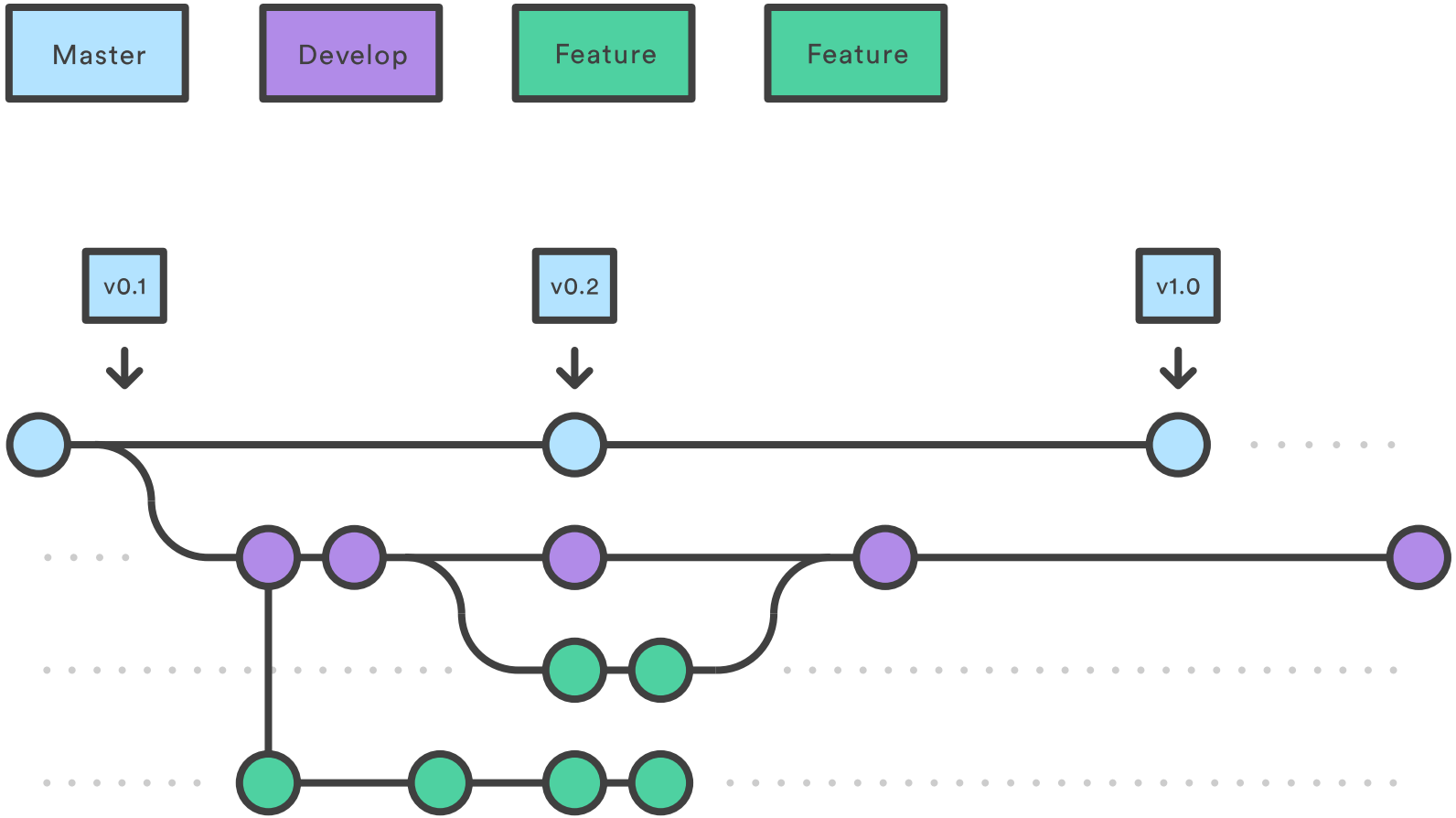
Git – o co chodzi z branchami (1)

https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow



Git – o co chodzi z branchami (2)

https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow



Redagowanie na GitHubie

UDOSTĘPNIAM DO REDAKCJI

- github.com: Settings > Branches > Branch protection rules > Add rule > master > Require pull request reviews before merging
- github.com: Settings > Manage access > Invite a collaborator
- *[tutaj ktoś redaguje moją dokumentację]*
- *Mail: dostaję zawiadomienie „requested your review”*
- github.com: przeglądam *pull request* do swego projektu i akceptuję zmiany, z którymi się zgadzam – inne komentuję albo odrzucam
- konto.github.io: oglądam swój projekt po zmianach

REDAGUJĘ

- *Mail: dostaję dostęp do projektu wybranej osoby*
- GitHub Desktop: clone
- GitHub Desktop: new branch
- *Visual Studio Code: redaguję pliki wg zasad STE*
- GitHub Desktop: commit, push, create pull request
- github.com: create pull request – dodaję właściciela projektu jako reviewera

Demo – 3 scenariusze

- Review z akceptacją wszystkiego
- Review z poprawkami
- Merge conflict

Git – branche

https://training.github.com/downloads/github-git-cheat-sheet/

ZADANIE	KOMENDA	GITHUB DESKTOP CLIENT
Stworzenie brancha	git branch [nazwa]	**Branch > New branch**
Przejście do brancha	git checkout [nazwa]	Current branch
Usunięcie brancha	git branch -d [nazwa]	Branch > Delete
Sprawdzenie jakie mamy branche i gdzie jesteśmy ☺	git branch	Current branch
Włączenie zmian z mojego brancha do main	git merge [branch]	**Branch > Create pull request**
Włączenie zmian z wybranego brancha do bieżącego	git merge [branch]	Branch > Merge into current branch

Ciąg dalszy ćwiczenia – termin 21.02.2021

Dokumentujemy po angielsku i publikujemy w GitHub Pages (*konto.github.io*)

1. **Instalację narzędzi – skąd Visual Studia Code, instalacja rozszerzeń**
2. **Opcje Visual Studio Code przydatne do pisania i generowania podglądu**
3. **Podstawy Markdowna – jakie funkcje są potrzebne, gdzie znaleźć opis**
4. **Stworzenie prywatnego repozytorium GitHub**
5. **Udostępnienie repozytorium wybranym użytkownikom**
6. **Stworzenie publicznego repozytorium GitHub Pages**
7. **Umieszczenie dokumentacji w repozytorium GitHub Pages**

- Publikowanie HTML z repozytorium za pomocą Jekylla (just-the-docs) – na 5
- Review online (branch, merge) – na 5
- *Dokumentacja zredagowana wg STE – na 5*

Przypomnienie:

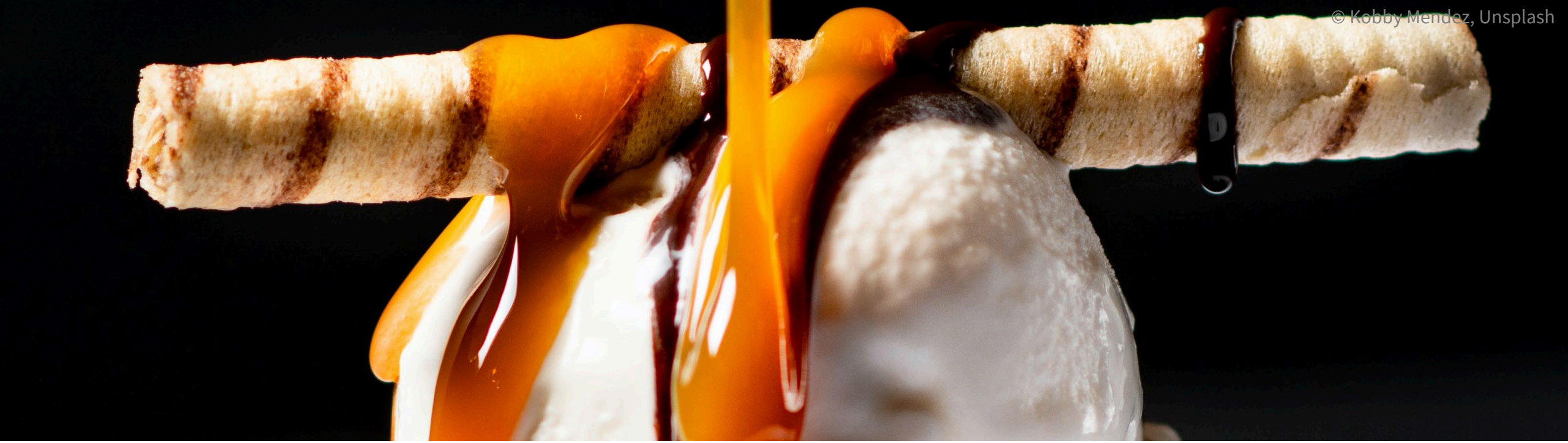
Analiza odbiorców i przypadków użycia – dokumentacja dla osoby, która nie była na zajęciach
Struktura dokumentacji – podział na tematy (np. VS Code, Markdown, GitHub, GitHub Pages)
Dokumentacja przygotowana jako konto.github.io może być ładnym dodatkiem do CV ☺

Pomysły na wykonanie ćwiczenia

- Będziemy sprawdzali dokumentację na *konto.github.io* - proszę zawiadomić nas mailem, że jest gotowa do oceny ☺
- Jeśli mamy pomóc w konfiguracji, to trzeba nadać nam dostęp
- Możecie robić redakcję wg STE w parach, jak na zajęciach
- Dokumentację z Mini Morrisa można usunąć ze swego konta lub zostawić jako materiał referencyjny w podrozdziale
- Można dodać odsyłacze do wideo na YT lub do ogólnodostępnych dokumentów

Prosimy o wypełnienie ankiety

https://forms.gle/P5aUi5zq33WQTWkt5



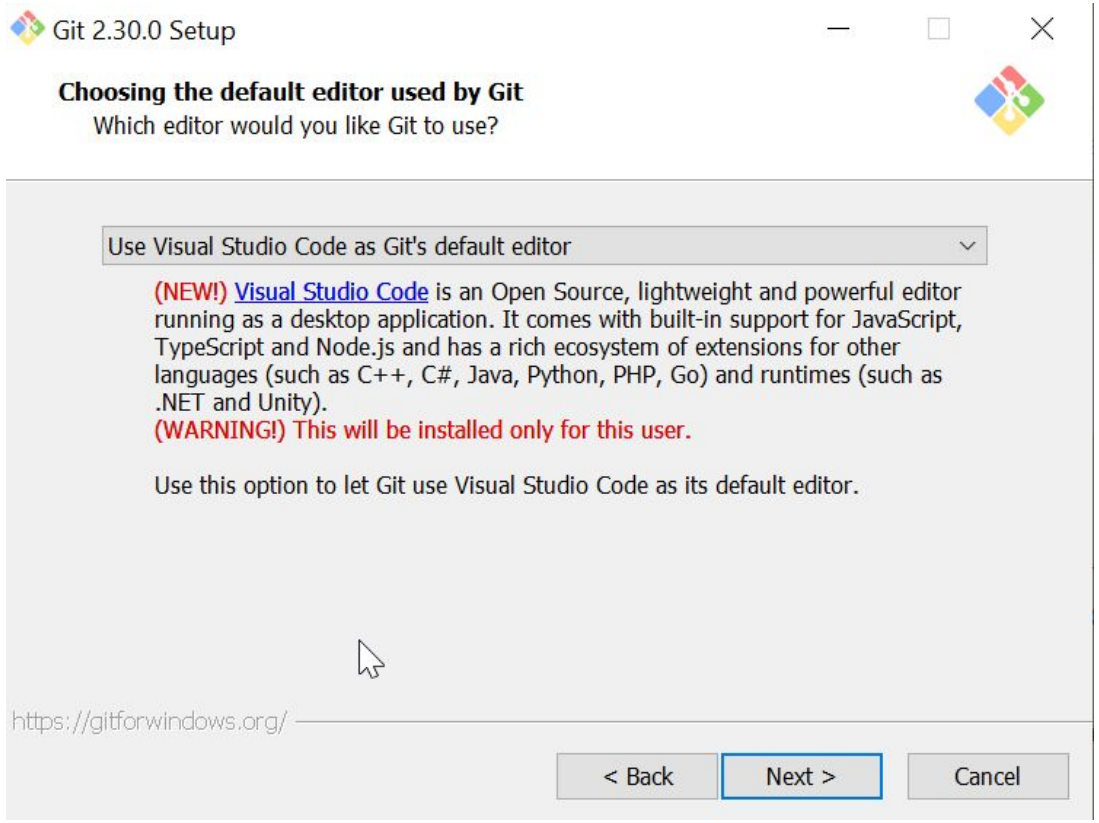
Na deser: licencje memoQ ☺

https://conflab.vistula.edu.pl/display/KOM/Licencje+memoQ+na+2+semestr

DODATKI

Visual Studio Code – instalacja gita

- Source Control
- *Nie ma gita* > install git
- https://git-scm.com
- Download 2.30.0
- *Instalacja*
- *Wybór edytora! Reszta domyślnie :)*
- *Ponownie otworzyć VS Code*
- **Open Folder**
  - Initialize Repository###
  - **Publish to GitHub**
- Authorize VS Code to GitHub
- Private repository
- *Pierwszy commit:*
  - git config –global user.email
  - git config –global user.name



Git – podstawowe funkcje

https://training.github.com/downloads/github-git-cheat-sheet/

ZADANIE	KOMENDA	VS CODE
Stworzenie projektu	git init [projekt]	Source Control > Initialize Repository
Wrzucenie projektu na serwer	git remote add origin [url]	Source Control > Publish to GitHub
Pobranie projektu z serwera	git clone [url]	More Actions > Clone
Dodanie pliku	git add [plik]	*Stworzenie pliku*
Usunięcie pliku	git rm [plik]	*Usunięcie pliku*
Zapisanie zmian	git stash	+ *(przy nazwie zmienionego pliku)*
Zatwierdzenie zmian	git commit -m"[sensowny opis]"	More Actions > Commit Staged
Wysłanie zmian na serwer	git push	More Actions > Push
Pobranie zmian z serwera	git pull	More Actions > Pull
Sprawdzenie jak się mamy ☺	git status	More Actions > Show Git Output
Pliki bez śledzenia przez gita	*Dodanie do .gitignore*	*Dodanie do .gitignore*

Git – branche (przyda się później ☺)

https://training.github.com/downloads/github-git-cheat-sheet/

ZADANIE	KOMENDA	VS CODE
Stworzenie brancha	git branch [nazwa]	
Przejście do brancha	git checkout [nazwa]	
Usunięcie brancha	git branch -d [nazwa]	
Sprawdzenie jakie mamy branche i gdzie jesteśmy ☺	git branch	
Włączenie zmian z wybranego brancha do bieżącego	git merge [branch]	