

Заняття 4.

1. HTML теги які ви маєте знати:

```
<html>, <head>, <body>, <title>, <h1>, <h2>, <h3>, <h4>, <h5>,
<h6>, <b>, <br>, <th>, <em>, <img>, <strong>,
<meta>, <p>, <a>, <i>, <ul>, <ol>, <li>, <dl>, <dt>, <dd>,

<table>, <tr>, <th>, <td>, <caption>, <thead>, <tbody>,
<tfoot>, <blockquote>

<q>, <map>, <area>, <sub>, <sup>, <cite>
```

2. Атрибути які ви маєте знати

```
charset,

src, title, alt, target, colspan, rowspan, width, height, href,
alt,

usemap, name, shape, coords
```

3. Git та GitHub

Git — розподілена система керування версіями файлів та спільної роботи. Є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні і злитті гілок. Для забезпечення цілісності історії та стійкості до змін заднім числом використовуються криптографічні методи, також можлива прив'язка цифрових підписів розробників до тегів і комітів.

<https://uk.wikipedia.org/wiki/Git>

GitHub — один з найбільших вебсервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc (раніше Logical Awesome).

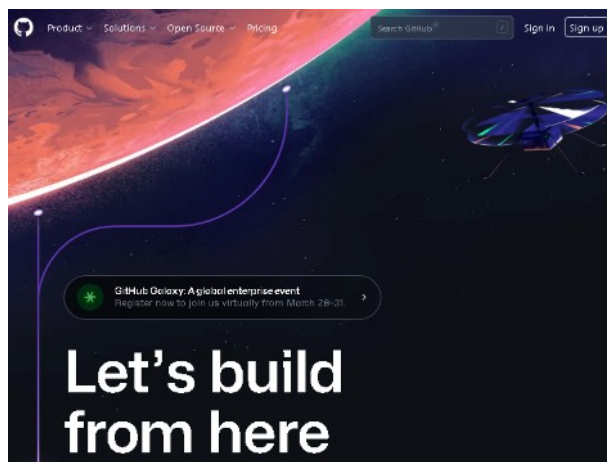
Сервіс безкоштовний для проєктів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проєктів пропонуються різні платні тарифні плани.

<https://uk.wikipedia.org/wiki/GitHub>

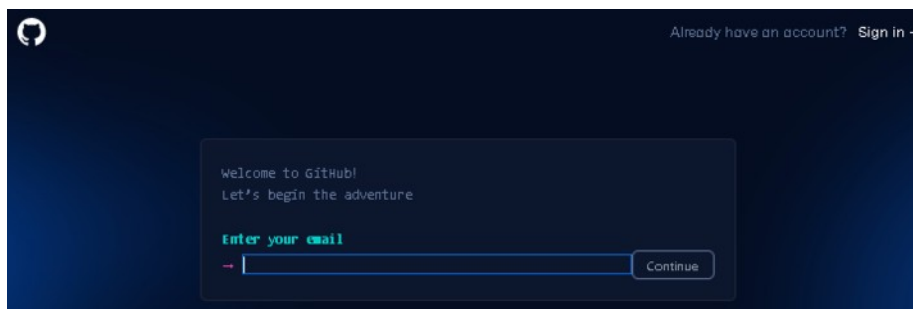
4. Завдання 1.

Реєстрація на GitHub.

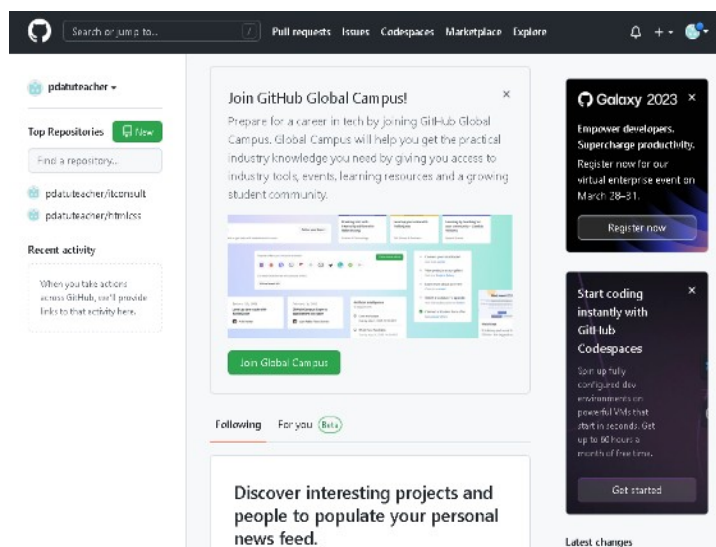
Перейдіть на сайт <https://github.com/>



Натиснувши на кнопку **[sing up]** розпочніть реєстрацію на сайті. Для цього вам знадобиться діюча електронна пошта.



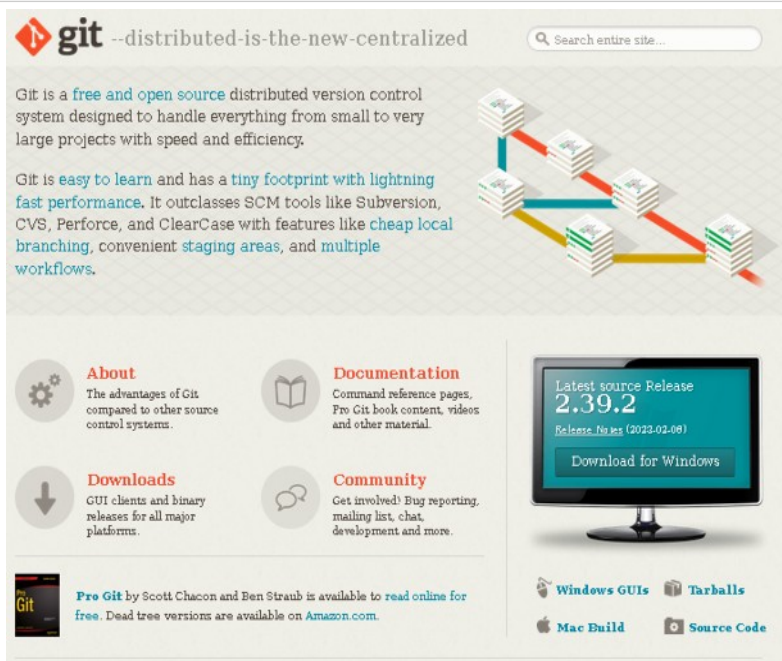
По закінченню реєстрації відкриється сторінка GitHub, де ви можете створювати власні репозиторії.



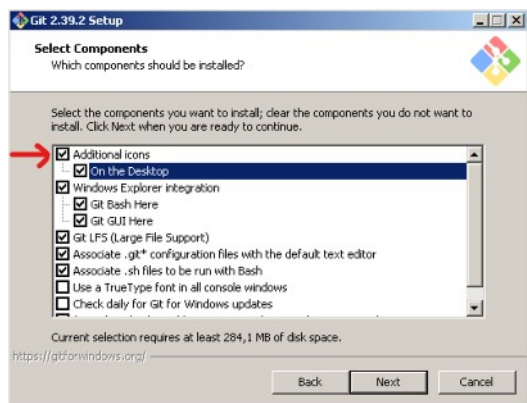
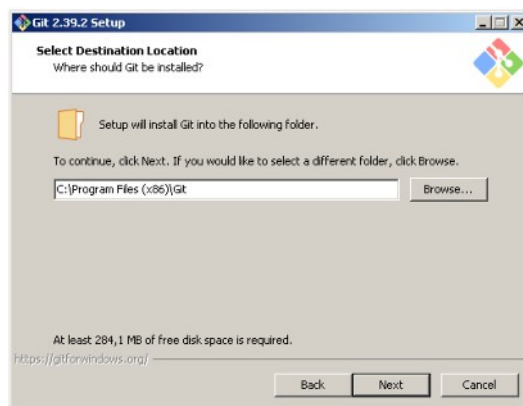
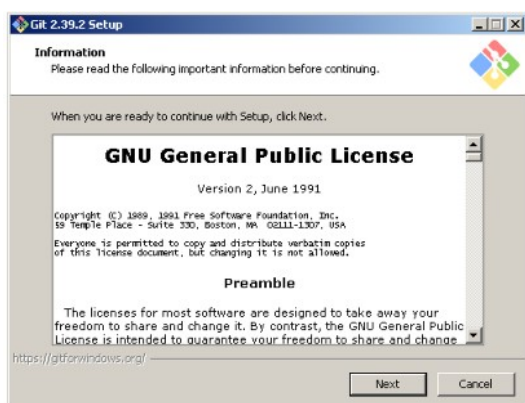
Домашнє завдання

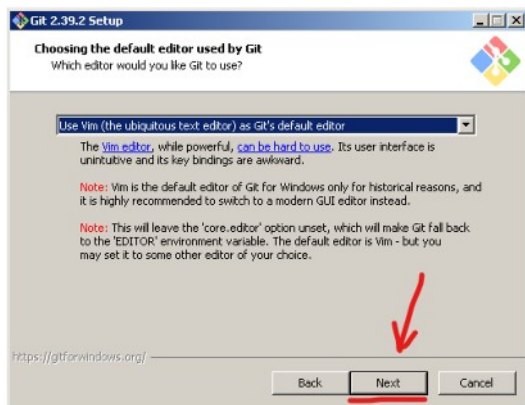
Увага!

Для роботи з GitHub вам знадобиться **Git**. Завантажити програму можна з цієї сторінки: <https://git-scm.com/>



Встановіть програму на свій комп'ютер



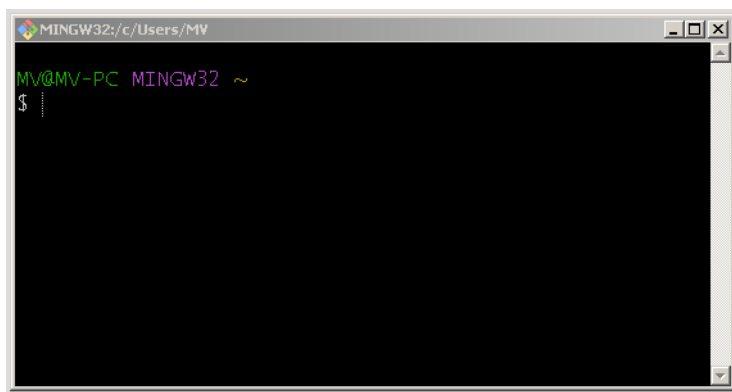


Далі достатньо натиснути **Next**, не змінюючи параметри.

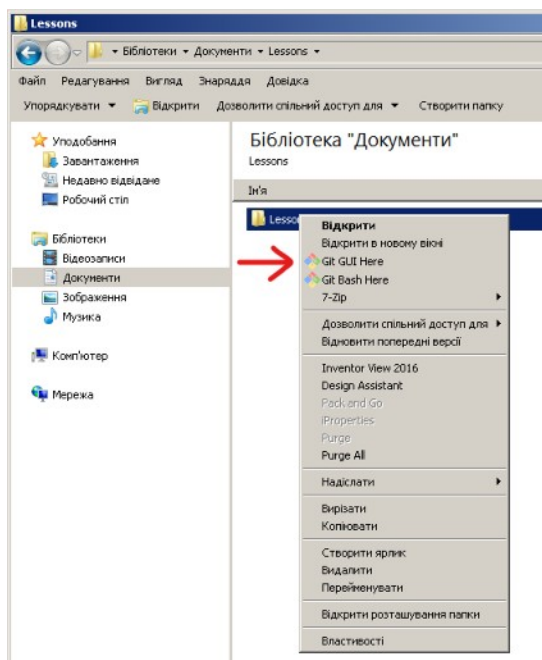
Якщо програма встановлена її можна запусити з ярлика на робочому столі, або через пункт меню "Пуск".



Це консольна програма, яка працює через командний рядок.

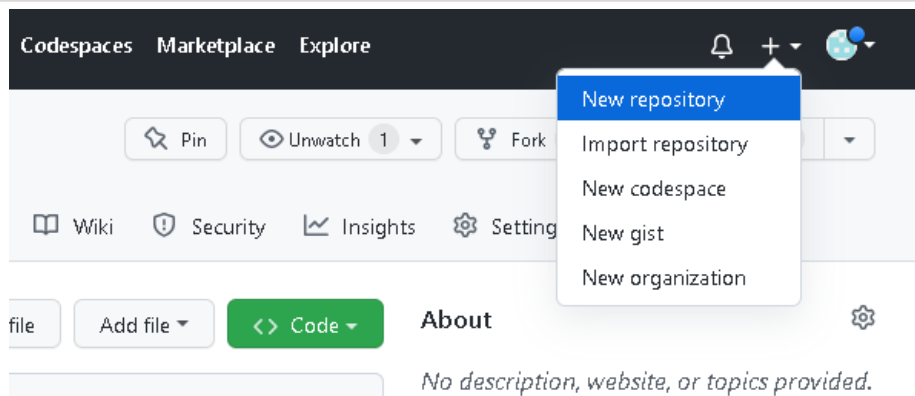


Програму можна запусити і за допомогою меню. Так навіть зручніше, тому що програма зразу відкривається в потрібній "папці".



5. Завдання 2.

Створити репозиторій на GitHub.



Введіть ім'я репозиторію. Це має бути "правильне" ім'я, тому що воно буде видиме усім користувачам інтернету. Залишіть репозиторій публічним.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

pdututeacher

Repository name *

htm1css



Great repository names are short and memorable. Need inspiration? How about [reimagined-octo-meme?](#)

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Вибравши відповідні для "Вас" пункти, натисніть кнопку **[Create repository]**

You are creating a public repository in your personal account.

Create repository

Після цього кроку, у вас відкриється вікно з командами для налаштування даного репозиторію.

pdatuteacher / htm1cssPublic

Pin

Unwatch1

Fork0

Star0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop

or

HTTPS

SSH

https://github.com/pdatuteacher/htm1css.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# htm1css" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/pdatuteacher/htm1css.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/pdatuteacher/htm1css.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Тепер ви можете працювати з даним репозиторієм, додаючи до нього файли.

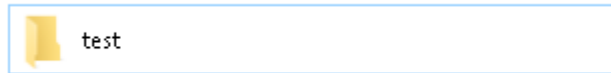
6. Деякі команди для роботи з GitHub

git init	Ініціалізувати порожній локальний Git-репозиторій
git status	Перевірити статус файлів
git add	Додати файл(и) в індекс <i>git add README.md</i> <i>git add .</i>
git commit	Покласти зміни в локальний репозиторій і одразу додати коментар <i>git commit -m "first commit"</i>
git branch -M main	Створити локальну гілку з назвою main <i>git branch -M main</i>
git remote add origin "URL"	Для роботи з віддаленим репозиторієм <i>git remote add origin https://github.com/pd...</i>
git push -u origin	Покласти зміни з локального у віддалений репозиторій <i>git push -u origin main</i>

7. Завдання 3.

Додати файли на віддалений репозиторій

1. В папці Мої документи створити нову папку **test**.



2. Відкрити програму Notepad++ і створити файл **index.html** в папці **test**.



3. За допомогою команди **git init** ініціалізувати порожній локальний Git-репозиторій.

```
$ git init
```

У вас з'явиться прихована папка **.git**



4. Відкрити файл **index.html** в редакторі і в тіло сторінки додати текст зі сторінки вашого репозиторію

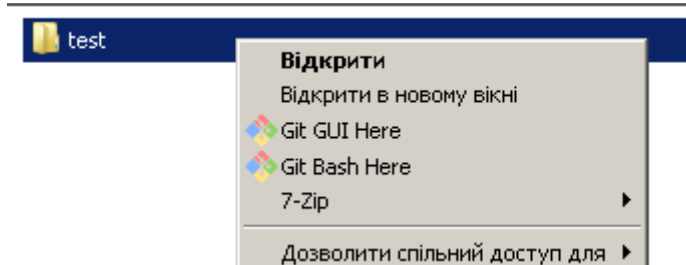
...or create a new repository on the command line

```
echo "# htmlcss" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/"Ваша адреса"
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/pdatuteacher/htmlcss.git
git branch -M main
git push -u origin main
```

5. Запустити програму **GitBash** в поточній папці.



6. Додати усі нові файли за допомогою команди **git add .**

Увага, після add, через пропуск стоїть крапка, це означає додати усі нові файли.

Перевірити будь які зміни можна за допомогою команди **git status**

```
$ git status
On branch master

No commits yet

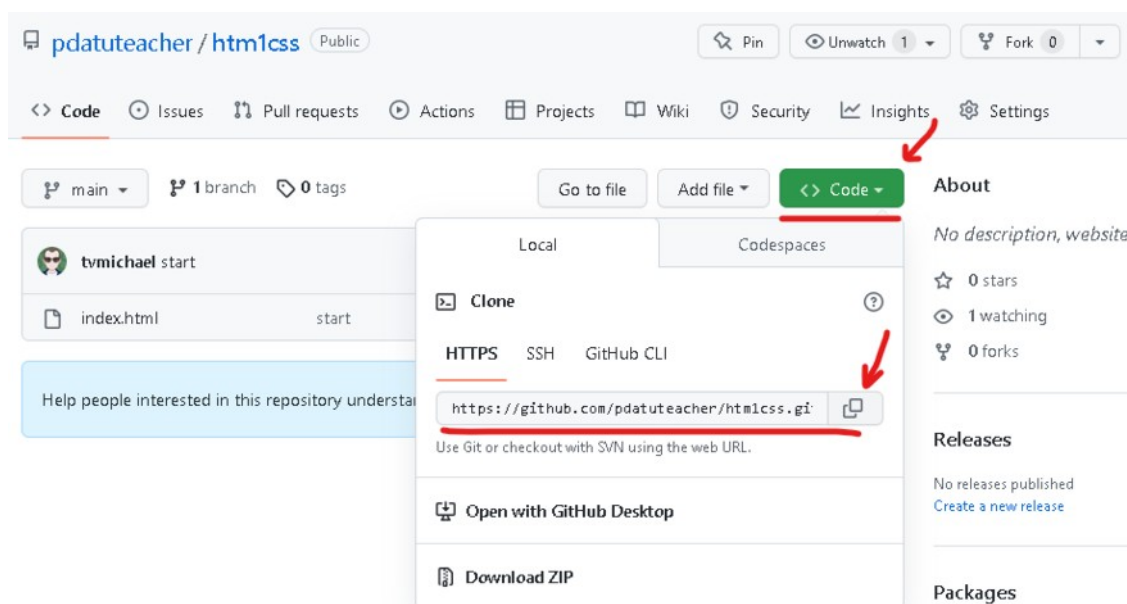
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html
```

7. Зробити *коміт* (зберегти зміни в історії Git). Це означає що ці зміни будуть збережені на вашому локальному комп'ютері, проте на віддаленому репозиторії їх ще нема.

```
$ git commit -m "add index"
[master (root-commit) c25ae6b] add index
1 file changed, 26 insertions(+)
create mode 100644 index.html
```

8. Вказати **URL** для роботи з віддаленим репозиторієм. Можна скопіювати зі сторінки на GitHub.

```
git remote add origin https://github.com/ "ваша адреса"
```



9. За допомогою команди **git branch -M main** створити локальну гілку з назвою main

```
$ git branch -M main
```

10.Зберегти зміни на віддаленому репозиторії за допомогою команди **git push -u origin main**

Після цієї команди система попросить вас ввести логін і пароль.

Увага! Дана послідовність команд буде працювати, у випадку якщо ви перший раз вносите зміни. Якщо до цього на віддаленому репозиторію вже були якісь зміни, або їх зробив хтось інший, треба буде виконати ще інші команди.