



JavaScript для початківців

Асинхронний JavaScript





Охріменко Дмитро
MCT



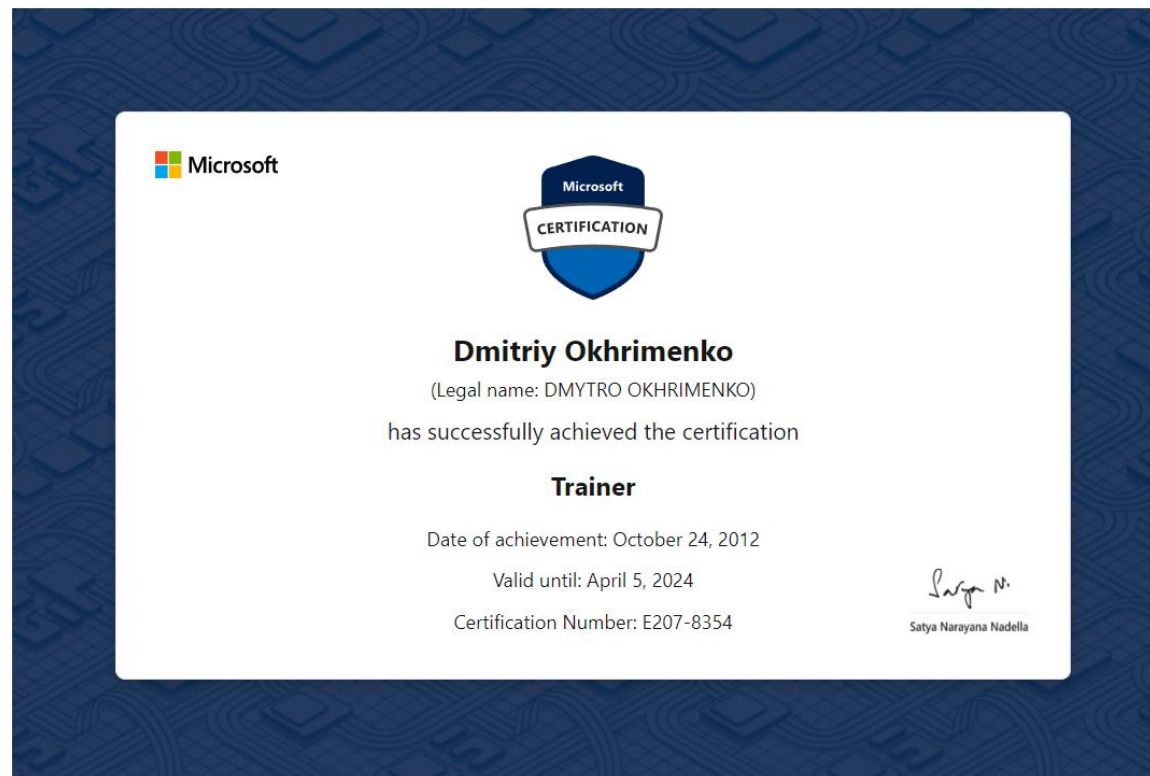
_okhrimenko



dmitriy.okhrimenko

co-founder
CyberBionic Systematics

co-founder
ITVDN

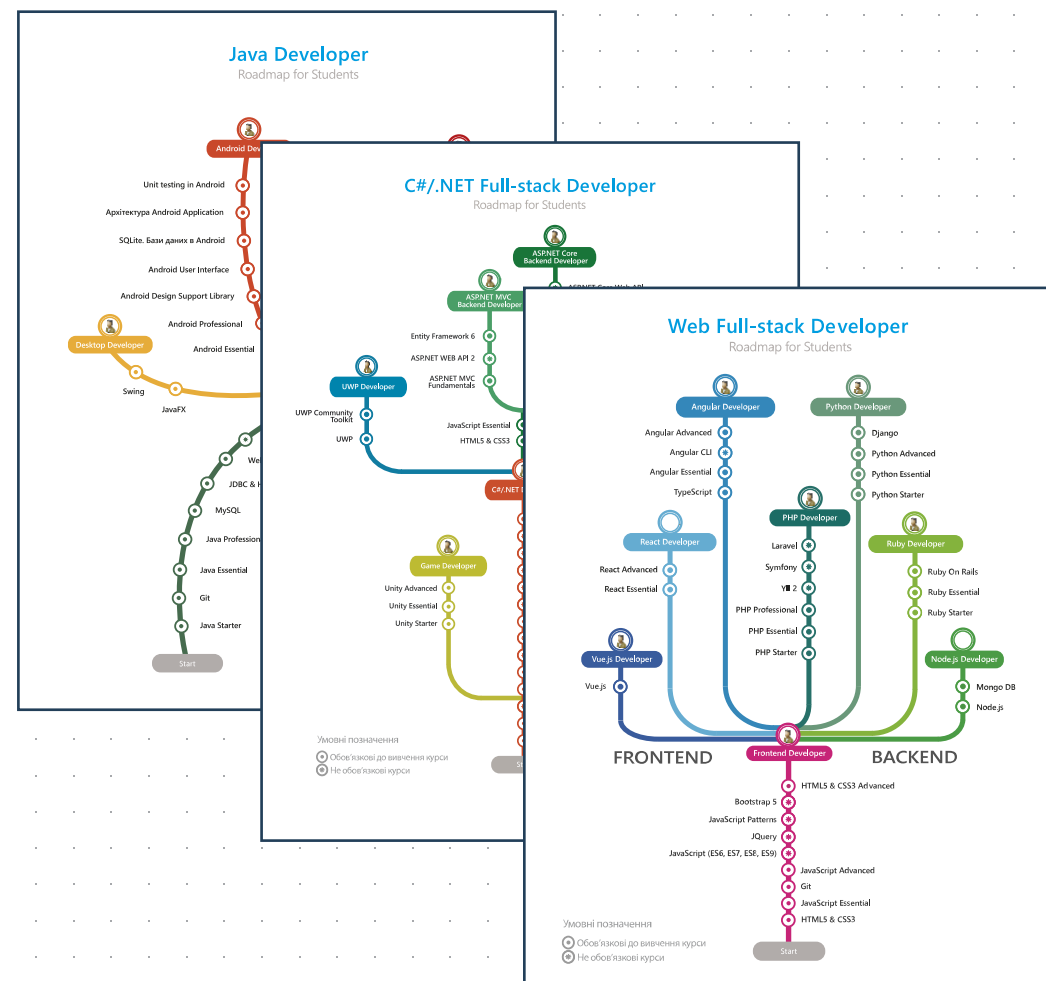


НАВЧАЛЬНИЙ ЦЕНТР CyberBionic Systematics

Живі онлайн лекції з тренером з
популярних ІТ напрямків

Навчання з подальшим стажуванням та
допомогою у працевлаштуванні

<https://edu.cbsystematics.com>





Відео курси з програмування

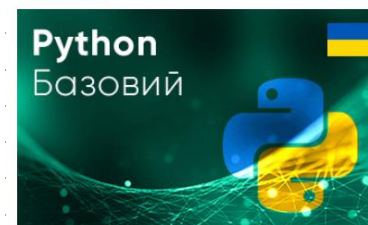
250+ відео курсів
20 спеціальностей

<https://itvdn.com>



Відео курси з веб розробки на
платформі itvdn

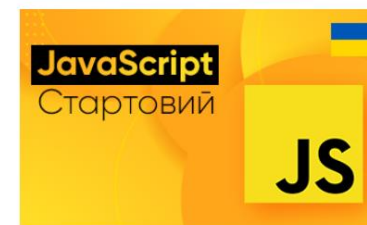
<https://itvdn.com/ru/catalog/specialities/frontend-developer>



Python Базовий
Вікторія Бойчук

Оновлено: 01.02.2023

⌚ 4 год 40 хв ▶ 9 уроків



JavaScript Стартовий
Дмитро Охріменко

Оновлено: 01.03.2023

⌚ 8 год 40 хв ▶ 14 уроків



HTML5 & CSS3 Стартовий
Віталій Мазяр

Оновлено: 10.11.2022

⌚ 6 год 24 хв ▶ 1



Python Стартовий
Руслан Скіра

Оновлено: 10.02.2023

⌚ 10 год 47 хв ▶ 10 уроків



Docker
Олег Заревич

Оновлено: 01.12.2022

⌚ 1 год 55 хв ▶ 5 уроків



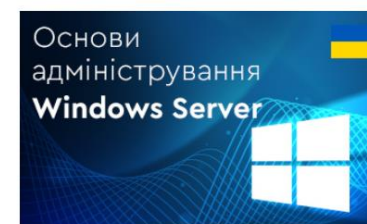
Основи використання Git
Дмитро Курган

Оновлено: 02.08.2023

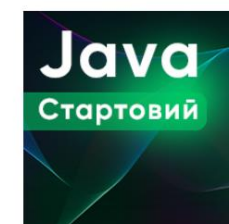
⌚ 2 год 15 хв ▶ 6



TypeScript
Ігор Михайличенко



Основи адміністрування Windows Server
Сергій Гапич



Java Стартовий
Дмитро Красний

План

Синхронний та асинхронний код
Структура HTTP запиту та відповіді
Fetch

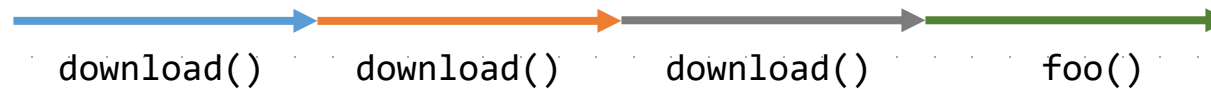
Синхронний код

Синхронний код - код, що виконується послідовно, кожна операція очікує на завершення попередньої.

```
function download() {  
    скачує та повертає дані  
    виконується ~10 секунд  
}
```

```
download(); ← ~10 сек.  
download(); ← ~10 сек.  
download(); ← ~10 сек.
```

```
// інші обчислення  
foo(); ← Виконається через ~30 сек.
```

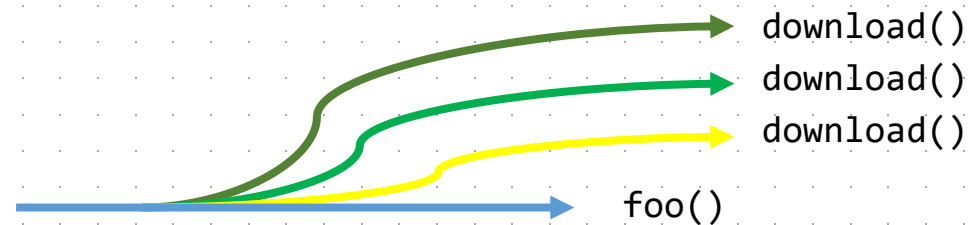


Асинхронний код

Асинхронний код – код, що виконується паралельно, а не послідовно. У більшості випадків асинхронне виконання коду передбачає виділення нових ресурсів для виконання асинхронних викликів - потоків.

```
download(); ← Не чекаємо завершення та переходимо до наступної операції.  
download(); ← Не чекаємо завершення та переходимо до наступної операції.  
download(); ← Не чекаємо завершення та переходимо до наступної операції.
```

```
// інші обчислення  
foo(); ← Виконається до того, як закінчиться виконання трьох методів
```

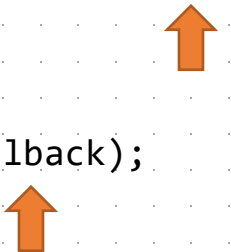


Організація асинхронного коду


Варіанти організації асинхронного коду:

- Callback (функція зворотного виклику)
- Promise
- Шаблон Observer


```
function callback(result) {  
  ...  
}  
  
download(callback);
```



```
let promise = download();  
  
promise.then(callback);
```



```
let observable = download();  
  
observable.subscribe(callback);
```



HTTP Запит

HTTP (HyperText Transfer Protocol) - протокол прикладного рівня передачі даних, спочатку - у вигляді гіпертекстових документів у форматі HTML. Нині використовується передачі довільних даних.

Структура запиту:

Стартовий рядок
Заголовки
Тіло запиту

```
1 GET http://google.com/ HTTP/1.1
2 Host: google.com
3 Connection: keep-alive
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
  q=0.9
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en,en-US;q=0.9,ru;q=0.8,uk;q=0.7
9 Cookie: SEARCH_SAMESITE=CgQIh5IB; HSID=AhGK-US0o7_t9Ista; APISID=uSb8jgmdX76IhRW7/A!
  SID=-Ad8XcgDyBI9W16vCXfM7-XcpzDR5MOOuDlnzDmXwrObfZQ4L5YxAtIooz8vqvm7e2v7gA.;
  SIDCC=Aji4QfEWEIeqG34J-dgPrs8CEKqftByjNb5VGHyNQmhuZLxPxoXoE9JFBQBDJtJHud9-9dL9_SDF
```



HTTP запит

Стартовий рядок визначає тип повідомлення.

Структура стартового рядка:

- Метод
- URI
- Версія

HTTP методи (HTTP дієслова):

GET	запит ресурсу
POST	відправлення сутності
PUT	зміна сутності
DELETE	видалення сутності
CONNECT	встановлює з'єднання
OPTIONS	опис параметрів з'єднання
HEAD	запит заголовків без тіла
PATCH	часткова зміна ресурсу

```
1 GET http://google.com/ HTTP/1.1
2 Host: google.com
3 Connection: keep-alive
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4398.96 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en,en-US;q=0.9,ru;q=0.8,uk;q=0.7
9 Cookie: SEARCH_SAMESITE=CgQIh5IB; HSID=AhGK-US0o7_t9Ista; AFSID=-Ad8XcgDyBI9W16vCXfM7-XcpzDR5MOOuDlnzDmXwrObfZQ4L5YxAtIc; SIDCC=Aji4QfEWEIeqG34J-dgPrs8CEKqftByjNb5VGHYnQmhuZLxPxoXoE9
```

HTTP Відповідь

Версія

Код стану

Заголовки

Тіло відповіді

Коди стану:

1xx інформаційний

2xx успіх

3xx редірект

4xx помилка клієнта

5xx помилка сервера

```
1 HTTP/1.1 200 OK
2 Date: Mon, 31 May 2021 10:37:27 GMT
3 Expires: -1
4 Cache-Control: private, max-age=0
5 Content-Type: text/html; charset=ISO-8859-1
6 P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
7 Server: gws
8 X-XSS-Protection: 0
9 X-Frame-Options: SAMEORIGIN
10 Set-Cookie: 1P_JAR=2021-05-31-10; expires=Wed, 30-Jun-2021 10:37:27 GMT; path=/; domain=.google.com; Secure
11 Set-Cookie:
  NID=216=EH_-b1KN8dDQu-RbS6eG4jJn7rJhViK20_TFL8JV-DSJGQJH1598JcbWxTWI3yJuHEufWs5j-gyUBYEXIwPq2pYfmCmqFInJdh1YLxm_BUzvU3JUn-OFmq1S365_JG1Hc7ch
  A1BQx7nNdIzdjQHuzcskUVFtiXk8b61iE6NUJA; expires=Tue, 30-Nov-2021 10:37:27 GMT; path=/; domain=.google.com; HttpOnly
12 Accept-Ranges: none
13 Vary: Accept-Encoding
14 Transfer-Encoding: chunked
15
16 <!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="uk"><head><meta content="text/html; charset=UTF-8"
  http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</
  title><script nonce="s55qi7z4JSSa3pLwviBVQw==">(function(){window.google={kEI: 'Z7y0YNmQKo6SaMyCpaAK', kEXPI: '0,772215,1,530320,56873,954,
  5105,206,4804,2316,383,246,5,1354,5250,1122516,1197783,469,30,328985,51223,16109,19403,9287,17572,4859,1361,9291,3027,4740,12841,4998,13228,
```

101 – Switching Protocol

200 – OK

201 – Created

301 – Moved Permanently

302 – Found

400 – Bad Request

404 – Not Found

500 – Internal Server Error

501 – Not Implemented

fetch

AJAX (**A**synchronous **J**avaScript **a**nd **X**ML) – термін, що визначає мережевий запит, зроблений за допомогою JavaScript коду.

fetch() – сучасний метод для виконання мережного запиту у JavaScript коді.

```
let promise = fetch (url, [options]);
```

url – URL для надсилання запиту

options – об'єкт, який описує додаткові параметри: HTTP метод, заголовки тощо.