

# Урок по програмиране за деца - Правим приложение за времето!

## Урок 1: Какво е програмиране?

### Какво е програмирането?

**Програмирането** е като да даваш инструкции на компютъра. Инструкцията е команда, която казва на компютъра какво да направи.

Представи си, че компютърът е робот, който не знае нищо. Робот е машина, която изпълнява команди.

Ти трябва да му кажеш **ТОЧНО** какво да прави, стъпка по стъпка. Стъпка е едно действие в поредица от действия.

### Пример от ежедневието:

Как да си направиш сандвич? 

1. Вземи хляб
2. Отрежи две филийки
3. Намажи масло
4. Сложи кашкавал
5. Залепи филийките

Това са **инструкции**! В програмирането правим същото, но за компютъра.

## Урок 2: Нашият проект - Weather Station

### Какво е Weather Station?

**Weather Station** означава "Метеорологична станция" на английски. Метеорологична станция е място, където се измерва времето.

Нашето **приложение** показва времето в различни държави. Приложение е програма, която работи на компютър или телефон.

### От какво се състои приложението?

Приложението има две основни части:

1. **Frontend** (фронтенд) - това е "лицето" на приложението. Frontend е частта, която виждаш на екрана.
2. **Backend** (бекенд) - това е "мозъкът" на приложението. Backend е частта, която не виждаш, но тя прави всички изчисления.

## Урок 3: Frontend - Красивата част

### Какво е HTML?

**HTML** означава HyperText Markup Language. Това е език за създаване на уеб страници.

HTML е като **скелет** на човек - дава структура. Структура означава как са подредени нещата.

```
html

<div class="weather-app">
  <h1>Времето в София</h1>
  <p>Слънчево</p>
</div>
```

Тук:

- `<div>` е като кутия. Кутия е контейнер, в който слагаме неща.
- `<h1>` е голямо заглавие. Заглавие е текст, който е по-важен от другия текст.
- `<p>` е параграф. Параграф е обикновен текст.

### Какво е CSS?

**CSS** означава Cascading Style Sheets. Това е език за красота на уеб страниците.

CSS е като **дрехите и боичките** - прави нещата красиви. Стил означава как изглежда нещо.

```
css

.weather-app {
  background-color: white; /* Фонът е бял */
  border-radius: 10px; /* Ъглите са заоблени */
  box-shadow: 0 4px 8px gray; /* Има сянка */
}
```

Тук:

- `background-color` е цветът на фона. Фон е това, което е зад текста.

- `border-radius` прави ъглите заоблени. Заоблен означава не остър, а плавен.
- `box-shadow` добавя сянка. Сянка е тъмната част, която пада от предмет.

## ⚡ Какво е JavaScript?

**JavaScript** е език за програмиране, който прави страниците интерактивни. Интерактивен означава, че можеш да взаимодействаш с него.

JavaScript е като **вълшебна пръчка** - оживява нещата! Оживява означава кара нещата да се движат и променят.

```
javascript

function showWeather(country) {
  // Вземи информация за времето
  // Покажи я на екрана
}
```

Тук:

- `function` е функция. Функция е набор от инструкции с име.
- `showWeather` е името на функцията. Име е как наричаме нещо.
- `country` е параметър. Параметър е информация, която даваме на функцията.

## 📺 Урок 4: Backend - Умната част

### 💻 Какво е сървър?

**Сървър** е компютър, който "сервира" (дава) информация. Сервира означава да обслужваш, да даваш на други.

Представи си сървъра като **келнер в ресторант**:

- Ти (frontend) поръчваш пица 🍕
- Келнерът (server) отива в кухнята
- Донася ти пицата

### 📡 Какво е API?

**API** означава Application Programming Interface. Това е начин две програми да си говорят.

API е като **меню в ресторант** - показва какво можеш да поръчаш. Меню е списък с възможности.

В нашия проект:

GET /weather/bulgaria - Дава времето в България  
GET /weather/uk - Дава времето във Великобритания

Тук:

- `GET` означава "вземи". GET е команда за получаване на информация.
- `/weather/bulgaria` е адресът. Адрес е място, където можеш да намериш нещо.

## Урок 5: TypeScript и Angular

### Какво е TypeScript?

**TypeScript** е като по-умен JavaScript. По-умен означава, че има повече възможности.

TypeScript проверява за грешки преди да стартираш програмата. Грешка е когато нещо не е правилно.

```
typescript  
  
let temperature: number = 22; // Това е число  
let description: string = "Слънчево"; // Това е текст
```

Тук:

- `number` означава число. Число е 1, 2, 3, и т.н.
- `string` означава текст. Текст е думи и букви.
- `:` казва "това е от тип". Тип е вид на данните.

### Какво е Angular?

**Angular** е framework за правене на приложения. Framework е набор от готови инструменти.

Angular е като **LEGO конструктор** - имаш готови части, които сглобяваш. Сглобявам означава слагам частите заедно.







```
typescript  
  
@Component({  
  selector: 'app-root',  
  template: '<h1>Здравей!</h1>',  
})  
export class AppComponent {}
```

Тук:

- `@Component` е декоратор. Декоратор е специална бележка, която дава допълнителна информация.
- `selector` е името в HTML. Selector е начин да избереш елемент.
- `template` е HTML кодът. Template е шаблон, модел.

## Урок 6: Как работи Weather Station?

### Процесът стъпка по стъпка:

1. **Потребителят избира държава** 
  - Потребител е човекът, който използва приложението
  - Избира означава да посочиш една опция от много
2. **Frontend изпраща заявка** 
  - Заявка е молба за информация
  - Изпраща означава праща през интернет
3. **Backend получава заявката** 
  - Получава означава приема това, което е изпратено
  - Backend проверява каква информация искаш
4. **Backend намира данните** 
  - Данни са информация (числа, текст)
  - Намира означава търси и открива
5. **Backend връща отговор** 
  - Отговор е информацията, която си поискал
  - Връща означава изпраща обратно
6. **Frontend показва данните** 
  - Показва означава прави видимо на екрана
  - Данните се появяват красиво оформени

### Пример с код:

Frontend иска данни:

```
typescript
```

```
this.weatherService.getWeather('bulgaria').subscribe(data => {  
  this.weatherData = data;  
});
```

Тук:

- `weatherService` е услуга. Услуга е помощник, който прави нещо за теб.
- `getWeather` е метод. Метод е действие, което може да извърши услугата.
- `subscribe` означава "чакай отговор". Subscribe е като да се абонираш за вестник.

## Backend дава данни:

```
javascript  
  
app.get('/weather/:country', (req, res) => {  
  const data = weatherData[country];  
  res.json(data);  
});
```

Тук:

- `app.get` казва "когато някой иска да вземе". App е съкращение от application (приложение).
- `req` е request (заявка). Request е молбата за информация.
- `res` е response (отговор). Response е отговорът с информацията.

## Урок 7: Основни концепции

### Какво са променливите?

**Променлива** е като кутия с име, в която държиш неща. Променлива се нарича така, защото можеш да промениш съдържанието ѝ.

```
typescript  
  
let temperature = 22; // Кутия "temperature" с число 22  
let city = "София"; // Кутия "city" с текст "София"  
let isSunny = true; // Кутия "isSunny" с "да" (true)
```

Тук:

- `let` означава "направи кутия". Let е ключова дума за създаване на променлива.
- `=` означава "сложи вътре". Равно е знак за присвояване на стойност.

- `22`, `"София"`, `true` са стойности. Стойност е това, което слагаме в кутията.

## Какво са циклите?

**Цикъл** е повтаряне на действия. Цикъл е като да правиш нещо отново и отново.

typescript

```
for (let hour of forecast) {  
  console.log(hour.time + ": " + hour.temp);  
}
```

Тук:

- `for` означава "за всеки". For е ключова дума за цикъл.
- `of` означава "от". Of показва откъде взимаме елементите.
- Това ще покаже всеки час от прогнозата. Прогноза е предсказание за бъдещето.

## Какво са условията?

**Условие** е проверка дали нещо е вярно. Условие е като въпрос с отговор "да" или "не".

typescript

```
if (temperature > 25) {  
  console.log("Горещо е!");  
} else {  
  console.log("Приятно е!");  
}
```

Тук:

- `if` означава "ако". If е ключова дума за условие.
- `>` означава "по-голямо от". Това е знак за сравнение.
- `else` означава "иначе". Else се изпълнява когато условието не е вярно.

## Урок 8: Инструменти за програмиране

### Какво е VS Code?

**VS Code** е програма за писане на код. Код е текстът, който пишем за компютъра.

VS Code е като **Word за програмисти** - има специални функции за код. Функция е възможност на програмата.

## Какво е npm?

**npm** означава Node Package Manager. Това е магазин за готов код.

npm е като **магазин за LEGO части** - можеш да вземеш готови неща. Пакет е готов код, който някой друг е написал.

```
bash  
npm install express
```

Тук:

- `npm` е програмата магазин. NPM управлява пакетите.
- `install` означава "инсталирай". Инсталирай означава сложи в компютъра.
- `express` е пакет за правене на сървър. Express е име на популярен пакет.

## Какво е Git?

**Git** е програма за запазване на версии на кода. Версия е различно състояние на кода във времето.

Git е като **машина на времето за код** - можеш да се върнеш назад. История е записът на всички промени.

```
bash  
git add .  
git commit -m "Добавих нова функция"
```

Тук:

- `git add` означава "запомни тези промени". Add означава добави.
- `git commit` означава "запази завинаги". Commit е като снимка на кода.
- `-m` е за съобщение. Съобщение обяснява какво си променил.

## Урок 9: Практически съвети

### Как да намираме грешки?

**Дебъгване** е намиране и поправяне на грешки. Debug означава "махни буболечките" (bugs = буболечки).

#### 1. Чети съобщенията за грешки



- Съобщение за грешка казва какво не е наред
- Обикновено показва на кой ред е проблемът

## 2. Използвай `console.log()` 🖨️

- `Console.log` показва информация в конзолата
- Конзола е специален прозорец за програмисти

```
typescript
```

```
console.log("Температурата е: " + temperature);
```

## 3. Провери правописа ✎

- Често грешките са от сгрешени букви
- JavaScript е чувствителен към главни и малки букви

## 🎨 Как да правим красиви приложения?

### 1. Използвай цветовете смислено 🌈

- Топли цветове (червено, оранжево) за важни неща
- Студени цветове (синьо, зелено) за спокойни неща

### 2. Остави празно място 🟪

- `Padding` е вътрешно разстояние
- `Margin` е външно разстояние

### 3. Направи го **responsive** 📱

- `Responsive` означава да работи на всякакви екрани
- Използвай проценти вместо фиксирани размери

## 📁 Урок 10: Следващи стъпки

### 🚀 Какво можеш да добавиш към **Weather Station**?

#### 1. Повече държави 🌐

- Добави Германия, Франция, Испания
- Всяка държава е нов обект в данните

#### 2. Икони за времето 🌤️

- Слънце за слънчево ☀️
- Облак за облачно ☁️

- Капки за дъжд 🌧️

### 3. Анимации 🎬

- Анимация е движение на екрана
- CSS transitions правят плавни преходи

### 4. Запазване на любима държава 💾

- LocalStorage пази данни в браузъра
- Браузър е програмата за интернет (Chrome, Firefox)

## 📖 Какво да учиш след това?

### 1. Базы данни 🗄️

- База данни е място за постоянно съхранение на информация
- MongoDB, PostgreSQL са популярни бази данни

### 2. Автентикация 🔒

- Автентикация е проверка кой си
- Позволява на всеки потребител да има свой профил

### 3. Deployment 🌐

- Deployment е качване на приложението в интернет
- Всеки ще може да го използва

## 🎯 Последни съвети:

### 1. Грешките са нормални ✅

- Всеки програмист прави грешки
- От грешките се учим най-добре

### 2. Питай когато не разбираш ❓

- Няма глупави въпроси
- Програмистите си помагат един на друг

### 3. Практикувай всеки ден 📅

- Програмирането е като спорт
- Колкото повече тренираш, толкова по-добър ставаш

### 4. Забавлявай се! 🎉

- Програмирането е творчество
  - Можеш да създадеш каквото си поискаш!
-

## Поздравления!

Вече знаеш основите на уеб програмирането! Уеб програмиране е създаване на сайтове и приложения за интернет.

Продължавай да учиш, експериментирай и създавай! Експериментирай означава да пробваш нови неща.

Ти можеш да станеш страхотен програмист! 🌟

---

## Речник на термините:

- **API** - начин програмите да си говорят
- **Backend** - невидимата част, която прави изчисления
- **CSS** - език за красота на уеб страници
- **Frontend** - видимата част на екрана
- **HTML** - език за структура на уеб страници
- **JavaScript** - език за интерактивност
- **npm** - магазин за готов код
- **Server** - компютър, който дава информация
- **TypeScript** - по-умен JavaScript
- **Variable** - кутия за съхранение на данни