



PHP 0 – FORMS BASICS

СТОЙНОСТИ ПО ПОДРАЗБИРАНЕ

- оператора за съкращение ?? (Null Coalescing Operator) – проверява дали дадена стойност е нулева (или не е зададена) и, ако е, връща стойността по подразбиране

```
$credits = trim($_POST['credits']) ?? 3; // Ако $_POST['credits'] е нула или  
липсва, ще се зададе стойност 3
```

- условния оператор if, но това е по-дълъг подход

```
$credits = trim($_POST['credits']);
```

```
if (!$credits) {
```

```
    $credits = 3; // Стойност по подразбиране
```

```
}
```

СТОЙНОСТИ ПО ПОДРАЗБИРАНЕ

Заглавие: <?= isset(\$course['title']) ? \$course['title'] : 'Няма заглавие'; ?>

Преподавател: <?= isset(\$course['teacher']) ? \$course['teacher'] : 'Няма преподавател'; ?>

Описание: <?= isset(\$course['description']) ? \$course['description'] : 'Няма описание'; ?>

Група: <?= isset(\$course['group']) ? \$course['group'] : 'Няма група'; ?>

Кредити: <?= isset(\$course['credits']) ? \$course['credits'] : 'Няма кредити'; ?>

Заглавие: <?= \$course['title'] ?? 'Няма заглавие'; ?>

Преподавател: <?= \$course['teacher'] ?? 'Няма преподавател'; ?>

Описание: <?= \$course['description'] ?? 'Няма описание'; ?>

Група: <?= \$course['group'] ?? 'Няма група'; ?>

Кредити: <?= \$course['credits'] ?? 'Няма кредити'; ?>

ПРЕОБРАЗУВАНЕ НА СПИСЪК () В HTML ТАБЛИЦА


```
<table border="1">
  <thead>
    <tr>
      <th>Заглавие</th>
      <th>Преподавател</th>
      <th>Описание</th>
      <th>Група</th>
      <th>Кредити</th>
    </tr>
  </thead>
  <tbody>
    <?php if (!empty($_SESSION['courses1'])): ?>
    <?php foreach ($_SESSION['courses1'] as $course): ?>
      <tr>
        <td><?= $course['title'] ?? 'Няма заглавие'; ?></td>
        <td><?= $course['teacher'] ?? 'Няма преподавател'; ?></td>
        <td><?= $course['description'] ?? 'Няма описание'; ?></td>
        <td><?= $course['group'] ?? 'Няма група'; ?></td>
        <td><?= $course['credits'] ?? 'Няма кредити'; ?></td>
      </tr>
    <?php endforeach; ?>
    <?php else: ?>
      <tr>
        <td colspan="5">Няма добавени курсове.</td>
      </tr>
    <?php endif; ?>
  </tbody>
</table>
```

СТЪПКИ

1. Преобразувахме списъка `` в таблица с тагове `<table>`, `<thead>`, `<tbody>`, и `<tr>`, като добавихме заглавен ред за всяка от колоните.
2. Използвахме условния оператор `??` (Null Coalescing Operator), за да проверим дали полетата съществуват, и ако не, да покажем стойност по подразбиране.
3. Ако няма добавени курсове, добавяме един ред, който информира потребителя за това. Това ще структурира информацията по по-ясен и подреден начин в таблица, което е подходящо за таблични данни като курсове и техните атрибути.

ЛОГИКА ЗА ИЗТРИВАНЕ НА КУРС

1. Ще добавим нова колона за изтриване

```
<th>Действия</th> <!-- Нова колона за действия -->
<!-- Бутон за изтриване, изпращащ POST заявка с индекса на курса -->
<form method="post" action="">
  <input type="hidden" name="delete_course" value="<?= $index ?>">
  <button type="submit">  Изтрий</button> <!-- Можем да използваме и друга иконка или текст -->
</form>
```

2. Добавяне на PHP логиката за изтриване на курс:

```
// Логика за изтриване на курс
if (isset($_POST['delete_course'])) {
    $index = (int) $_POST['delete_course']; // Получаваме индекса на курса

    if (isset($_SESSION['courses1'][$index])) {
        // Премахваме курса от масива
        unset($_SESSION['courses1'][$index]);

        // Пренареждаме масива след изтриване, за да няма пропуснати индекси
        $_SESSION['courses1'] = array_values($_SESSION['courses1']);
    }
}
```

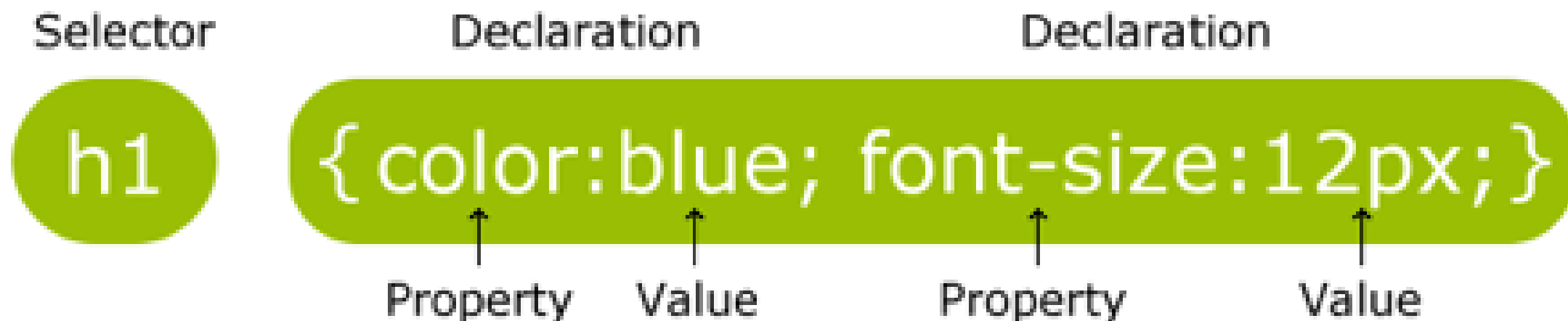
CSS - CASCADING STYLE SHEETS

- HTML описва съдържание сържанието, CSS описва, CSS описва визуализацията. Тоест:
 - Как да са подредени нещата на екрана.
 - Кое колко да е голямо.
 - Какъв цвят да е даден текст, фон и тн.

CSS ни позволява лесно да променяме външния вид на уеб страница. Първо избираме елементите, които искаме да стилизираме, използвайки селектори (като заглавия, параграфи или класове). След това задаваме стойности за определени свойства (като цвят или размер на шрифта), които искаме да променим. Браузърът прилага тези стилове на всички елементи, които отговарят на зададения селектор.

CSS Syntax

A CSS rule has two main parts: a selector, and one or more declarations:



The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

ИНЛАЙН CSS

```
<p style="color: red;">Текст 1</p>  
<p style="color: red;">Текст 2</p>  
<p style="color: red;">Текст 3</p>
```

STYLE TAG HTML

```
<style>  
  p.error {  
    color: red;  
  }  
</style>  
<p class="error">Текст 1</p>  
<p class="error">Текст 2</p>  
<p class="error">Текст 3</p>
```

FILE LINK

```
<link href="style.css" rel="stylesheet">
```

```
.error {  
  color: red;  
}
```

BAD PRACTICE

- **Трудна поддръжка:** Промените в стиловете трябва да се правят на много места, което е времеемко.
- **Нарушено разделение на съдържание и стил:** Смесването на HTML и CSS прави кода по-труден за четене и управление.
- **Трудно повторно използване:** Няма лесен начин за повторно използване на стилове за множество елементи.
- **Ниска производителност:** Браузърите трябва да анализират всеки инлайн стил поотделно, което може да забави зареждането на страницата.
- **Приоритет:** Инлайн стиловете имат по-висок приоритет и могат да създадат проблеми при промени чрез външен CSS файл.
- *Забележка: Инлайн стиловете могат да бъдат удобни за бързи промени или тестове, но за по-големи и поддържани проекти е много по-добре да се използва външен или вътрешен CSS. Това прави кода по-чист, по-гъвкав и по-лесен за поддръжка в дългосрочен план.*