

서버 사이트 스크립트 언어, PHP

PHP는 무엇이고, 활용 사례에는 어떤 것이 있을까?

1995: PHP is dead, learn ColdFusion

2002: PHP is dead, learn ASP.net

2003: PHP is dead, learn Django

2004: PHP is dead, learn Ruby on Rails

2010: PHP is dead, learn Flask

2011: PHP is dead, learn AngularJS

2016: PHP is dead, learn Next.js

2022: PHP is dead, learn Python

2023:



PHP는 죽지 않는다.

왜 자꾸 죽었대

<?php
echo "Hello, world!";
?>

인터프리터 방식



Rasmus Lerdorf



동적 웹 페이지를 만들기 위해 설계

전통적인 방식: PHP, HTML 혼용

```
🗇 코드 복사
php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <?php
       echo "<h1>Hello, World!</h1>";
    ?>
</body>
</html>
```

현대적인 방식: PHP, HTML 분리

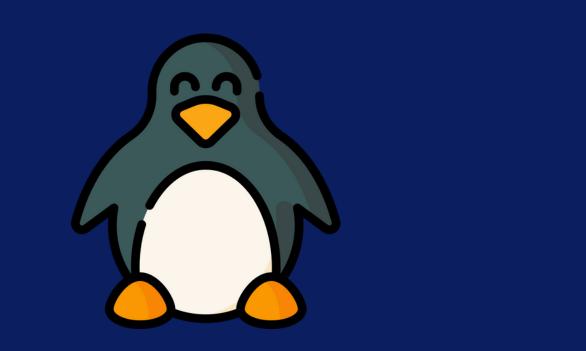
```
php
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class PageController extends Controller
    public function show()
        $message = "Hello, World!";
        return view('welcome', compact('message'));
```

```
blade
<!-- resources/views/welcome.blade.php -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <h1>{{ $message }}</h1>
</body>
</html>
```

PHP 특징

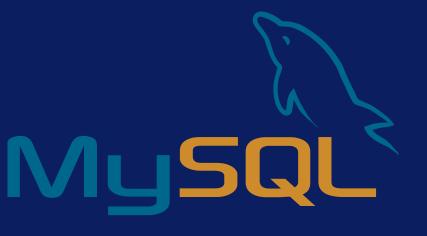
- MySQL, mSQL, Oracle 등 여러 DB 편리하게 연동
- 거의 모든 운영 체제에 구현 가능
- C/C++ 언어 문법과 비슷하고 클래스 지원
- 코드 작성이 쉽고 문법이 간단한 편
- ASP나 Perl 언어에 비해 처리 속도가 빠름
- 동적 이미지 생성 가능

최강 조합









Linux + Apache + PHP + MySQL

PHP 전망

PHP 8.0에서는 JIT 컴파일러가 도입되었고, 점진적인 타입 시스템이 도입되는 중

-> 지속적으로 발전하고 살아남을듯