

1. 기본 구문

- PHP 태그: `<?php ... ?>`
- 주석: `// 한 줄 주석, # 한 줄 주석, /* 여러 줄 주석 */`
- 출력: `echo "Hello"; print "World";`
- 변수: `$` 기호로 시작. \$다음에 문자나 밑줄로 시작. `$name = "Jules";`
- 문자열 연결: . 연산자 사용. `echo "Hello, " . $name;`
- 문자열 보간: 큰따옴표 안에서 변수 사용. `echo "Hello, $name";`

2. 변수 타입 및 연산자

- 타입: String, Integer, Float, Boolean, Array, Object, NULL.
- 배열:
 - 인덱스 배열: `$cars = array("Volvo", "BMW");` 또는 `$cars = ["Volvo", "BMW"];`
 - 연관 배열: `$ages = array("Peter"⇒35, "Ben"⇒37);` 또는 `$ages = ["Peter"⇒35];`
- 연산자: 산술, 할당, 비교, 논리 연산자는 다른 언어와 유사.
- 삼항 연산자: `(condition) ? true_expr : false_expr;`
- Null 병합 연산자 (PHP 7+): `$username = $_GET['user'] ?? 'nobody';`

3. 제어 흐름

- if-elseif-else:


```
if (condition) {
    // ...
} elseif (condition) {
    // ...
} else {
    // ...
}
```
- switch:


```
switch ($variable) {
    case "value1":
        // ...
        break;
    default:
        // ...
}
```
- while / do-while 루프.
- for 루프: `for ($i = 0; $i < 5; $i++) { ... }`

- foreach 루프:


```
$colors = ["red", "green", "blue"];
foreach ($colors as $value) {
    echo "$value\n";
}
```

4. 함수

- 함수 정의:


```
function greet($name = "Guest") {
    return "Hello, $name!";
}
echo greet("Jules");
```
- 타입 힌팅 (PHP 7+): `function add(int $a, int $b): int { ... }`

5. 클래스와 객체 (OOP)

- 클래스 정의:


```
class Car {
    // 프로퍼티
    public $color;

    // 생성자 (PHP 8+ 속성 승격)
    public function __construct(public string $model) {}

    // 메서드
    public function drive() {
        return "Driving a $this->model car!";
    }
}
```
- 객체 생성: `$myCar = new Car("Tesla");`
- 멤버 접근: `$myCar->drive();`
- 상속: `class ElectricCar extends Car { ... }`
- 인터페이스: `interface Drivable { ... }`
- 트레이트 (Traits): `trait Sharable { ... }`

6. 슈퍼글로벌 (Superglobals)

- 어디서든 항상 접근 가능한 내장 변수.
- `$_GET`: URL 파라미터로 전달된 데이터.
- `$_POST`: HTTP POST 요청으로 전달된 데이터.
- `$_REQUEST`: `$_GET`, `$_POST`, `$_COOKIE`의 내용을 포함.
- `$_SERVER`: 서버 및 실행 환경 정보.
- `$_SESSION`: 세션 변수.

- `$_COOKIE`: 쿠키 변수.
- `$_FILES`: 파일 업로드 정보.

7. 파일 처리

- 파일 읽기: `$content = file_get_contents("file.txt");`
- 파일 쓰기: `file_put_contents("file.txt", "New content");`

8. 데이터베이스 연동 (PDO)

PHP Data Objects (PDO)는 다양한 데이터베이스에 일관된 방식으로 접근하게 해주는 인터페이스입니다.

```
$host = '127.0.0.1';
$db = 'test';
$user = 'root';
$pass = '';
$charset = 'utf8mb4';

$dns =
"mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE            =>
PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE =>
PDO::FETCH_ASSOC,
];

try {
    $pdo = new PDO($dns, $user, $pass,
$options);
} catch (\PDOException $e) {
    throw new \PDOException($e-
getMessage(), (int)$e->getCode());
}

// 쿼리 실행
$stmt = $pdo->query('SELECT name FROM
users');
while ($row = $stmt->fetch()) {
    echo $row['name'] . "\n";
}

// Prepared Statement
$stmt = $pdo->prepare('SELECT * FROM
users WHERE email = ?');
$stmt->execute([$email]);
$user = $stmt->fetch();
```

9. Composer

PHP의 의존성 관리 도구.

- `composer.json`: 프로젝트의 종속성을 정의하는 파일.
- `composer install`: `composer.json`에 명시된 모든 종속성 설치.
- `composer update`: 종속성을 최신 버전으로 업데이트.
- `composer require vendor/package`: 새로운 종속성 추가.
- Autoloading: `require 'vendor/autoload.php';`