

🔗موضوع: معرفی توابع ناشناس (لَمبدا) و refactoring کد برای ایجاد انعطاف بیشتر

🔍بخش‌های مهم جلسه

1. چرا سرانغ *Lambda* می‌رویم؟

- توابعی که در جلسات قبلی ساختیم (مثل `filterByAuthor`) عملکردی دارند اما انعطاف‌پذیری‌شان محدود است. اگر بخواهیم شرط‌های پیچیده‌تر مثل «کتاب‌های بعد از سال ۲۰۰۰» یا موارد فیلتر بیشتر را پیاده کنیم، باید مرتباً تابع جدید بنویسیم که سخت و تکراری می‌شود.

2. معرفی *Lambda* یا *Anonymous Function*

- تابع بی‌نام:

```
$filterByAuthor = function ($books) {  
    // filtering logic  
};
```

- این امکان را می‌دهد که تابع را داخل یک متغیر ذخیره کرده و بعد فراخوانی کنیم:

```
$filtered = $filterByAuthor($books);
```

3. Refactor کردن تابع اولیه (*filterByAuthor*)

- برای انعطاف بیشتر تابع را طوری refactor می‌کنیم که به‌جای یک نویسنده، بتوانی کلید و مقدار موردنظر را فیلتر کنی:

```
function filter(array $items, string $key, $value) {  
    $filteredItems = [];  
    foreach ($items as $item) {  
        if ($item[$key] === $value) {  
            $filteredItems[] = $item;  
        }  
    }  
}
```

```

    }
    return $filteredItems;
}

```

- با این تابع می‌توان:

```

$weir = filter($books, 'author', 'Andy Weir');
$recent = filter($books, 'releaseYear', 2021);
```:contentReference[oaicite:8]{index=8}

```

4. مرحله بعد: استفاده از Callback برای انعطاف تمام

- برای فیلترینگ‌های پیچیده، تابع `filter()` را به شکل انعطاف‌پذیرتری با callback بازنویسی می‌کنیم:

```

function filter(array $items, callable $callback) {
 $filtered = [];
 foreach ($items as $item) {
 if ($callback($item)) {
 $filtered[] = $item;
 }
 }
 return $filtered;
}

```

- حالا می‌توان هر شرط دلخواهی نوشت:

```

$modern = filter($books, function($book) {
 return $book['releaseYear'] >= 2000;
});
```:contentReference[oaicite:11]{index=11}

```

5. استفاده از تابع داخلی `array_filter`

- این تابع داخلی PHP کار فیلترینگ را ساده‌تر می‌کند:

```

$weirBooks = array_filter($books, function($book) {
    return $book['author'] === 'Andy Weir';
});
```:contentReference[oaicite:13]{index=13}

```

---

## □ تمرین پیشنهادی

- تابع `filter()` با `callback` تعریف کن.
- کتاب‌های پس از سال ۲۰۱۰ یا نویسنده خاص را با آن فیلتر کن.
- یا از `array_filter` استفاده کن.

---

## ✓ جمع‌بندی دقیق:

- **Lambda functions** یا توابع ناشناس، انعطاف و قدرت بیشتری در فیلترینگ داده می‌دهند.
- **Refactor** کردن کمک می‌کند کد ساده‌تر و `scalable` باشد.
- **Callback** و `array_filter` راه حرفه‌ای برای مدیریت و پردازش داده در PHP هستند.