

PHP for Beginners - Session 9

Lambda Functions

In this session, we learn about anonymous functions (lambdas) and refactoring filters in PHP.

1. Why Lambdas?

- Traditional functions like `filterByAuthor` work but are not flexible.
- Lambdas allow us to pass custom filtering logic dynamically.

2. Anonymous Function Example:

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

3. Refactored Generic Filter Function:

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

Usage:

```
$weir = filter($books, 'author', 'Andy Weir');  
$recent = filter($books, 'releaseYear', 2021);
```

4. Callback-Based Filter Function:

```
function filter(array $items, callable $callback) {
```

PHP for Beginners - Session 9

```
$filtered = [];  
foreach ($items as $item) {  
    if ($callback($item)) {  
        $filtered[] = $item;  
    }  
}  
return $filtered;  
}  
  
$modern = filter($books, function($book) {  
    return $book['releaseYear'] >= 2000;  
});
```

5. Using array_filter (PHP built-in):

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

6. Summary:

- Lambda functions increase flexibility.
- Callback design pattern enables powerful filtering logic.
- array_filter is a native PHP helper for these tasks.

7. Exercise:

- Implement filter() with a callback.
- Filter books by year or author.