Hvordan bli en bedre programmerer

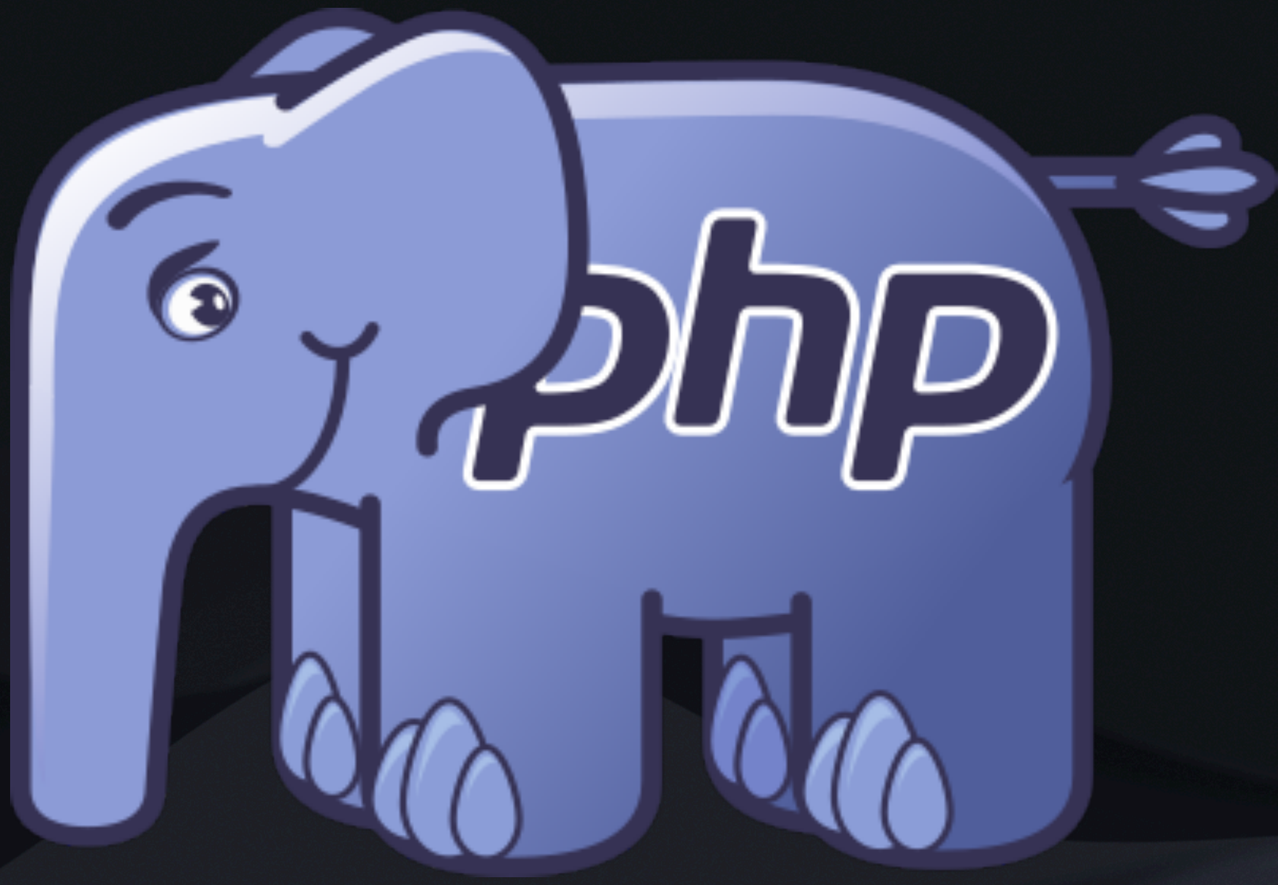
PHP Bergen 2024-01

@steinmb - 31.01.2024

History 1994-2024

PHP growing up

- Personal Home Page Tools - 1994
- FI
- Personal Home Page Construction Kit (1995)
- PHP/FI (1996)
- PHP 3 (1998)
- PHP 8.4 (Nov. 2024)



WIDE

```
<!--include /text/header.html-->
<!--getenv HTTP_USER_AGENT-->
<!--if substr $exec_result Mozilla-->
  Hey, you are using Netscape!<p>
<!--endif-->
<!--sql database select * from table where user='$username'-->
<!--ifless $numentries 1-->
  Sorry, that record does not exist<p>
<!--endif exit-->
  Welcome <!--$user-->!<p>
  You have <!--$index:0--> credits left in your account.<p>
<!--include /text/footer.html-->
```

```php
$itemPrices = array_map(
    fn ($orderLine) => $orderLine->item->price,
    $order->lines
);
```

Excerpt From
Front Line PHP 8.1 Brent Roose

7.0	3 December 2015[113]	10 January 2019[55]	Zend Engine 3 (performance improvements[53] and 64-bit integer support on Windows[114]), uniform variable syntax,[57] AST-based compilation process,[115] added `Closure::call()`,[116] bitwise shift consistency across platforms,[117] `??` (null coalesce) operator,[118] Unicode code point escape syntax,[119] return type declarations,[68] scalar type (integer, float, string and boolean) declarations,[69] `<=>` "spaceship" three-way comparison operator,[120] generator delegation,[121] anonymous classes,[122] simpler and more consistently available CSPRNG API,[123] replacement of many remaining internal PHP "errors" with the more modern exceptions,[56] and shorthand syntax for importing multiple items from a namespace.[124]
7.1	1 December 2016	1 December 2019[107]	`iterable` type,[125] nullable types,[126] `void` return type,[127] class constant visibility modifiers,[128] short list syntax,[129] multi-catch[130]
7.2	30 November 2017	30 November 2020[107]	Object parameter and return type declaration,[131] libsodium extension,[132] abstract method overriding,[133] parameter type widening[134]
7.3	6 December 2018[135]	6 December 2021	Flexible Heredoc and Nowdoc syntax,[136] support for reference assignment and array deconstruction with `list()`,[137] PCRE2 support,[138] `hrtime` function[139]
7.4	28 November 2019[140]	28 November 2022	Typed properties 2.0,[141] preloading,[142] null-coalescing assignment operator,[143] improve `openssl_random_pseudo_bytes`,[144] weak references,[82] foreign function interface (FFI),[145] always available hash extension,[146] password hash registry,[147] multibyte string splitting,[148] reflection for references,[149] unbundle ext/wddx,[150] new custom object serialization mechanism[151]
8.0	26 November 2020[152]	26 November 2023	Just-In-Time (JIT) compilation,[72] arrays starting with a negative index,[153] stricter/saner language semantics (validation for abstract trait methods),[154] saner string to number comparisons,[155] saner numeric strings,[156] `TypeError` on invalid arithmetic/bitwise operators,[157] reclassification of various engine errors,[158] consistent type errors for internal functions,[159] fatal error for incompatible method signatures[160]), locale-independent float to string conversion,[161] variable syntax tweaks,[162] attributes,[163][164][165][166] named arguments,[167] match expression,[168] constructor property promotion,[169] union types,[170] `mixed` type,[171] static return type,[172] nullsafe operator,[81] non-capturing catches,[173] `throw` expression,[80] JSON extension is always available.[174]
8.1	25 November 2021[175]	25 November 2024	Explicit octal integer literal notation,[176] enumerations,[177] read-only properties,[178] first-class callable syntax,[179] `new` in initializers,[180] pure intersection types,[181] `never` return type,[182] `final` class constraints,[183] fibers[184]
8.2	8 December 2022[185]	8 December 2025	Readonly classes,[186] `null`, `false`, and `true` as stand-alone types,[187][188] locale-independent case conversion,[189] disjunctive normal form types,[190] constants in traits[191]
8.3	23 November 2023[192]	23 November 2026	Typed class constants,[193] dynamic class constant fetch,[194] `#[\Override]` attribute,[195] deep-cloning of read-only properties,[196] new `json_validate` function,[197] randomizer additions,[198] the command-line linter supports multiple files

Derick Rethans

Core Developer

Gina Peter Banyard

Core Developer

Ilija Tovilo

Core Developer

Jakub Zelenka

Core Developer

Máté Kocsis

Core Developer

Arnaud Le Blanc

Core Developer

David Carlier

Core Developer

James Titcumb

Core Developer

Saki Takamachi

Core Developer

Shivam Mathur

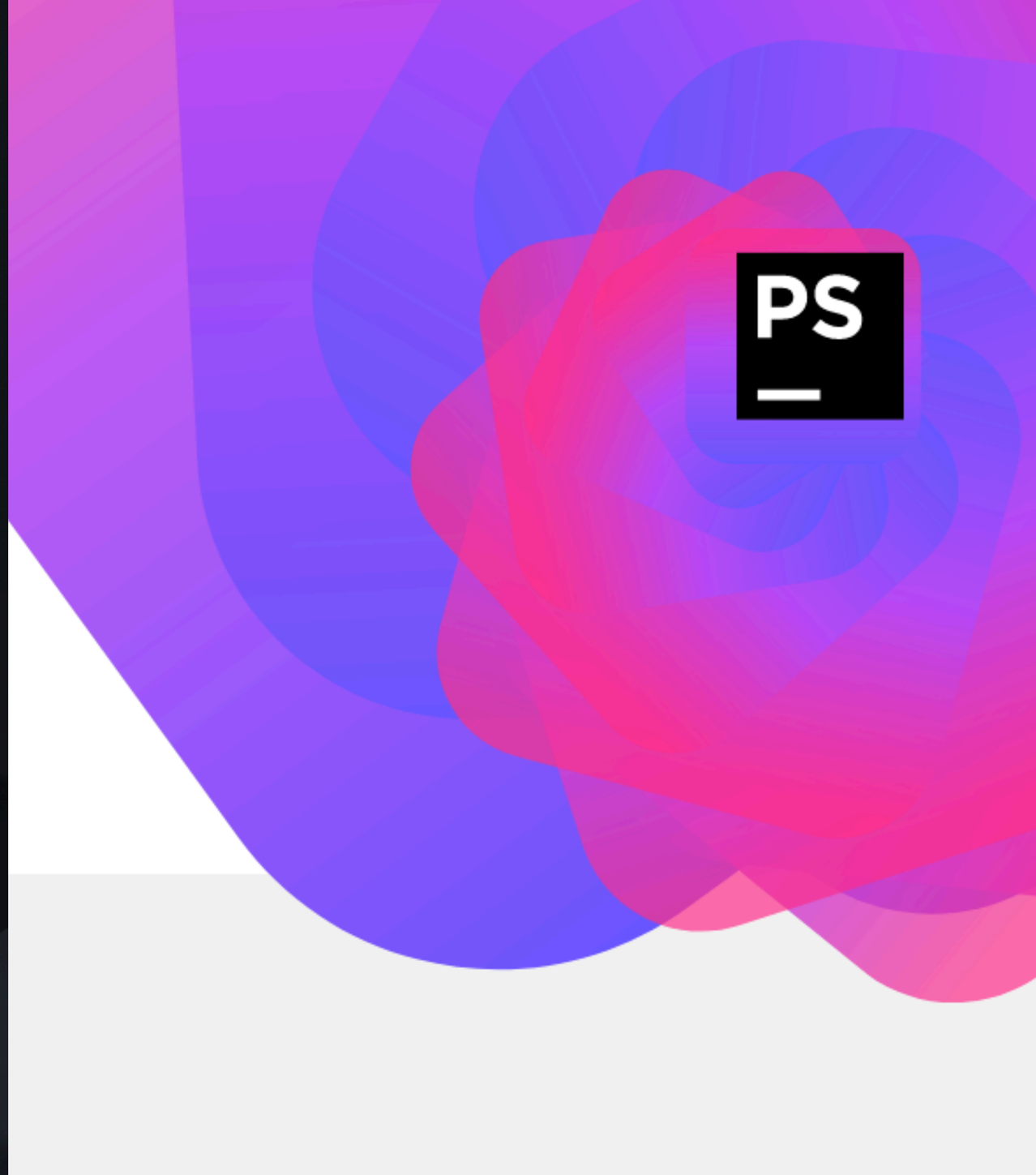
Core Developer

PHP growing up, are we?

Tools

PhpStorm

IDE from JetBrains



Xdebug

Go ahead, make my day

- Step Debugging
- Improvements to PHP's error reporting
- Profiling
- Code Coverage Analysis

Quality control

Static analysers

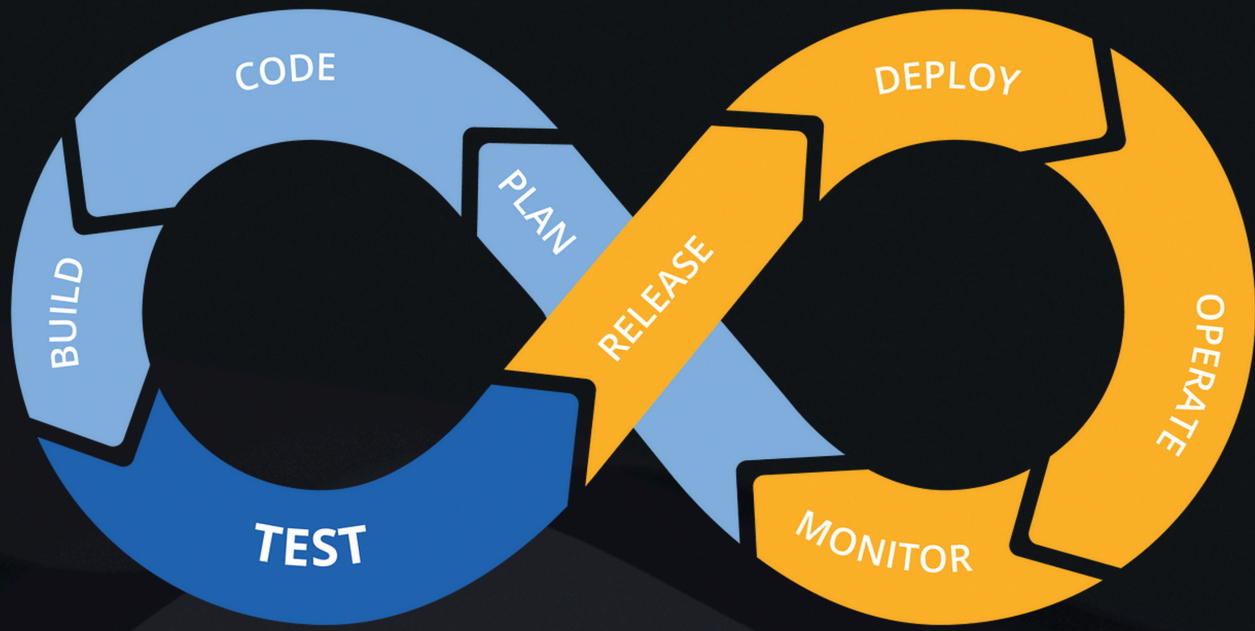
Code quality

- vimeo/psalm
- symplify/easy-coding-standard
- phpstan/phpstan
- phpcsstandards/php_codesniffer
- PhpStorm

CI/CD

Let others run your tests

- github.com
- gitlab.com
- <include *.com>

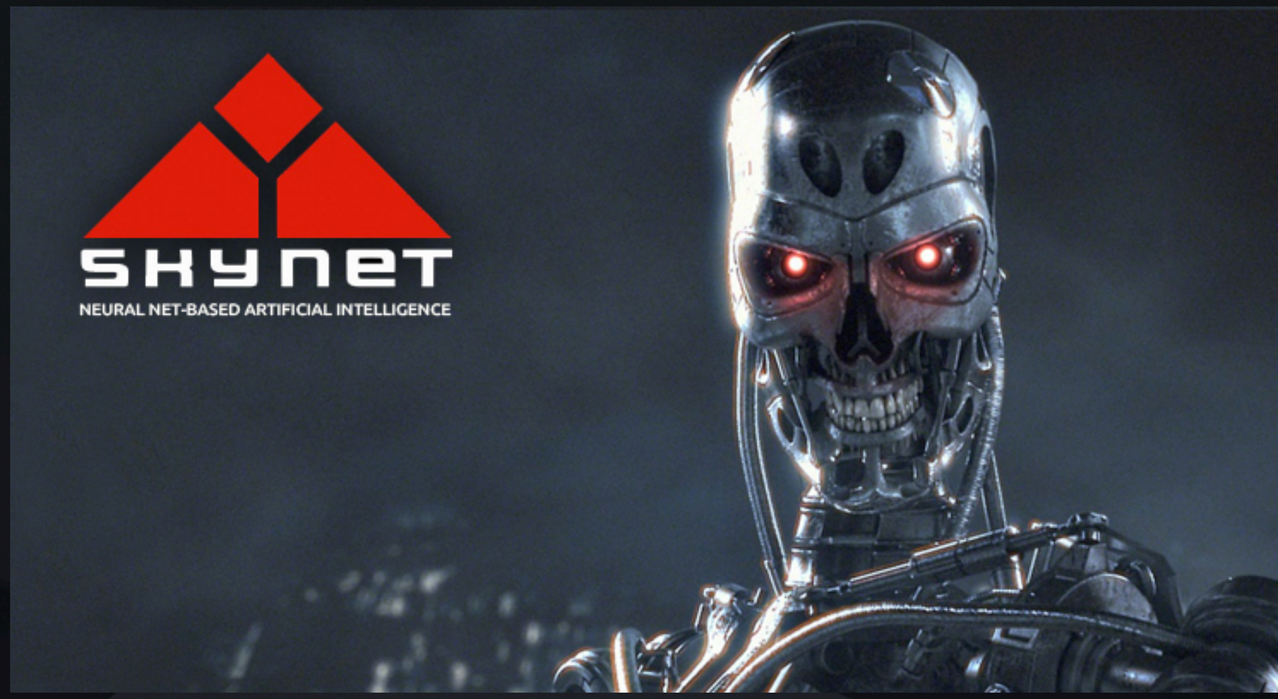


Get started with unit tests
Wed 28 Feb
https://www.meetup.com/php-bergen/events/298902700

AI

Chatdriven

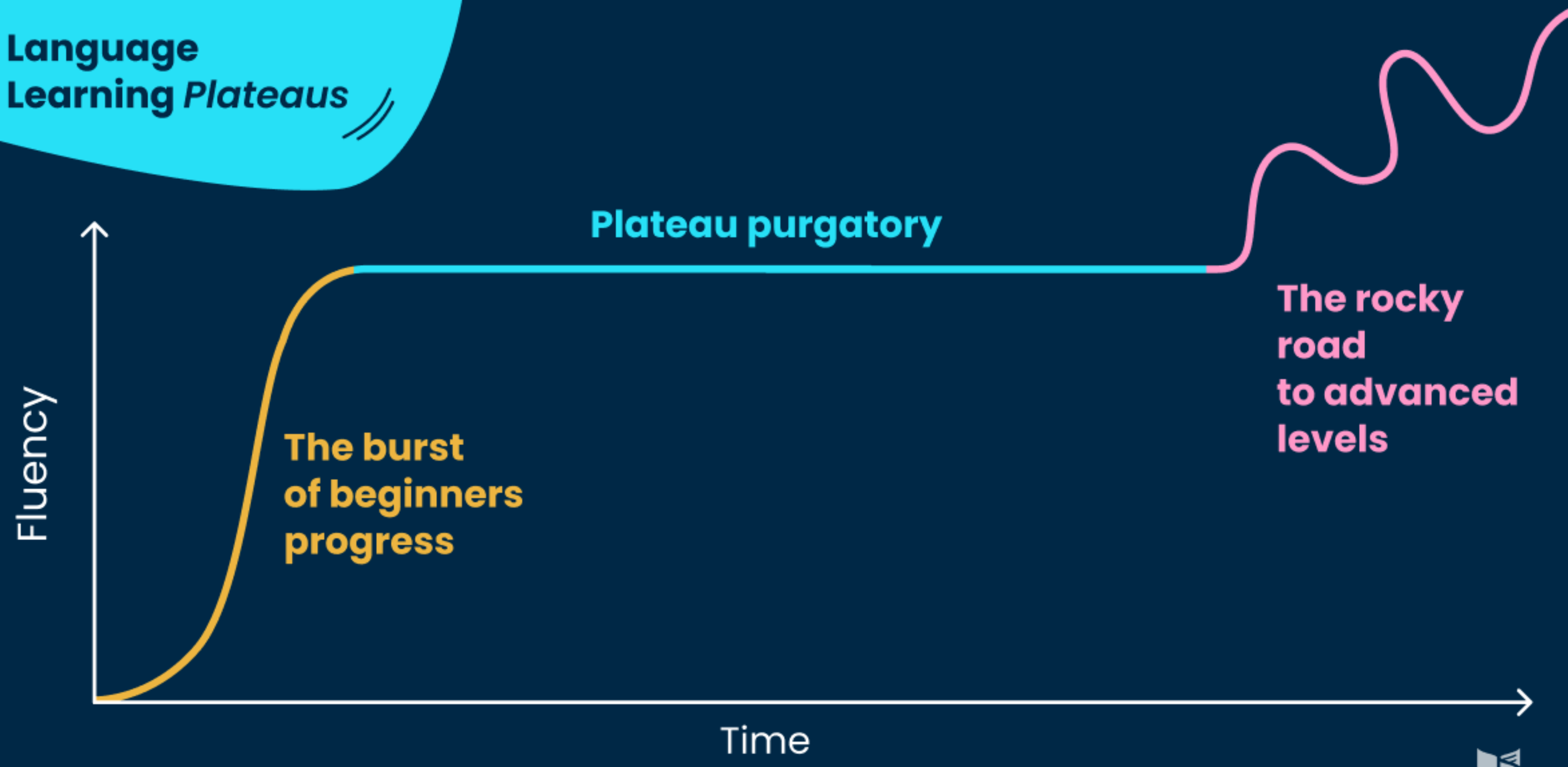
IDE driven



Awesome code === true?

Stagnation

Language Learning *Plateaus*



Why Do Language Learning Plateaus Occur?

This is what some linguistics say

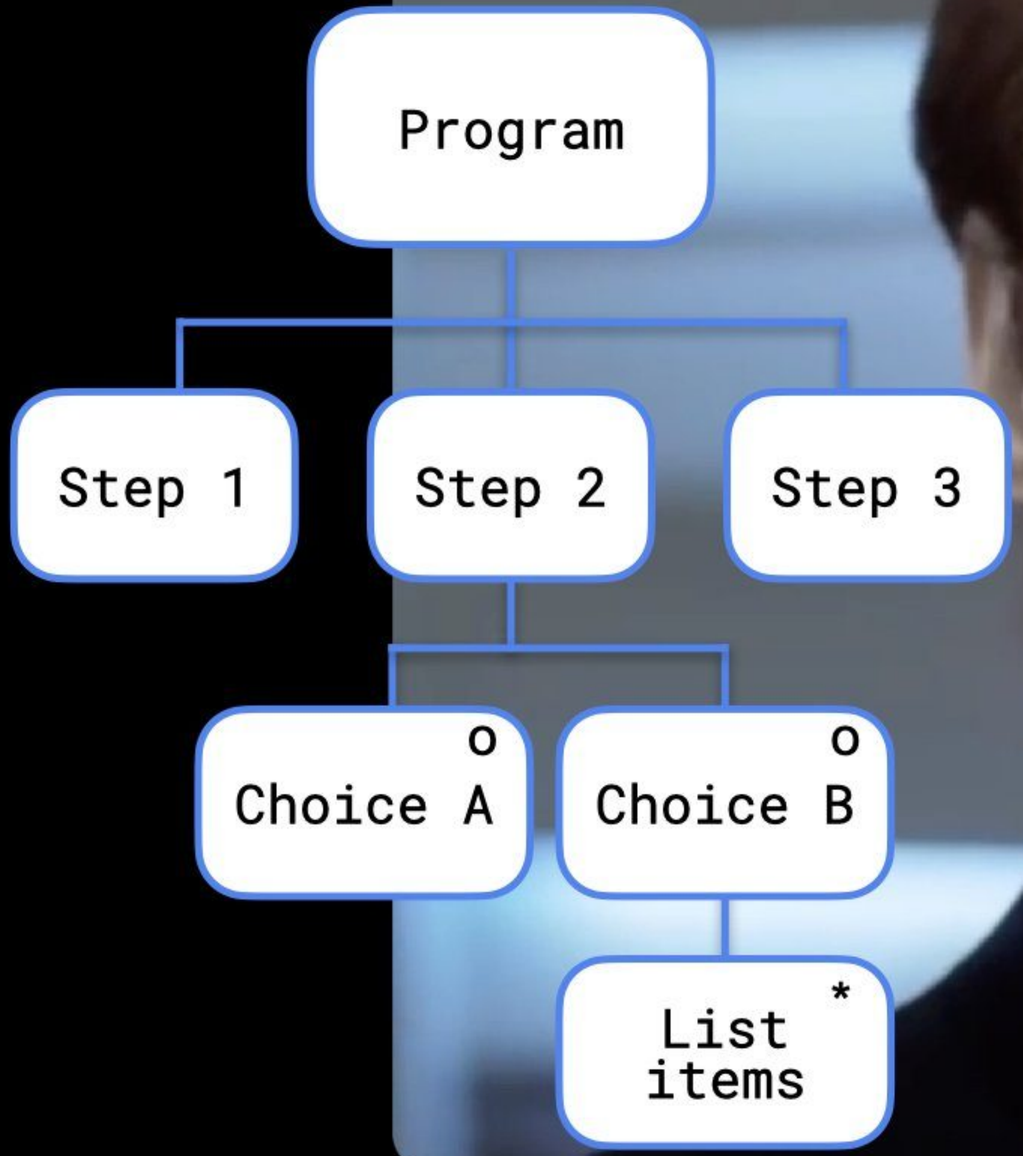
- You've met your goals.
- You don't have goals.
- You're stuck on a loop, doing the same thing all the time.
- There's a natural slowdown.

A Drupal story

Drupal 7 to 8

- Drupal 7 - Procedural code mostly
- Drupal 8 an increase of code made some where else.
- Composer driven
- Drupal 8 mostly OOP
- Forked

GOSUB and Subroutines in Commodore BASIC



File Edit Search Run Compile Debug Tools Options Window Help

PRNFLTR.PAS 1

```
const
  MaxAttributes = 8;

type
  TPCharArray = arra
  PPCharArray = ^TPC

  PPrinterCodes = ^T
  TPrinterCodes = re
      { Number of pr                                  }
    PreambleCount: B
      { Pointer to a                                  amble sequence for
        this printer
    Preamble: PPChar
      { Pointer to a                                  e sequences for
        changing the
    CodeArray: PPCha
      { Array of indexes into the CodeArray corresponing to attributes
        supported for this printer. }
    Attributes: array[0..MaxAttributes - 1] of Byte;
      { Codes sent at the start of a page. }
```

About

Turbo Pascal

Version 7.0

Copyright (c) 1983,92 by

Borland International, Inc.

OK

75:1

F1 Help | Welcome to Turbo Pascal. Press Enter to close this dialog box

DRUPALCON
AUSTIN
2014

Make the leap

It is time to learn something new.

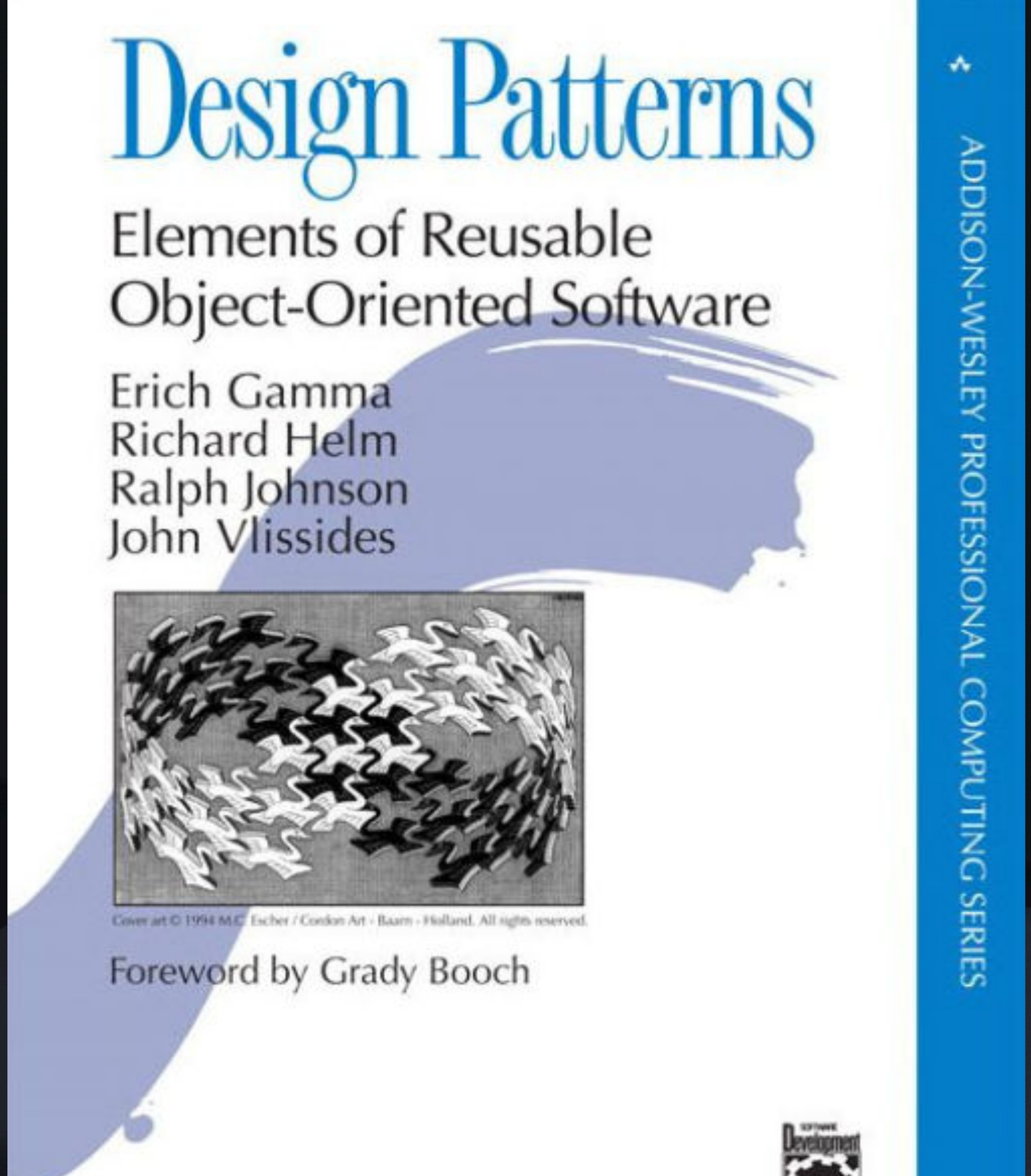
New features attracts developers from other languages

Expectations and patterns

Common way of describing design patterns in programming

Design Patterns: Elements of Reusable Object-Oriented Software

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides



Resources

Social

Talk to humans and humanoid Cylon

- PHP Bergen
- Present at PHP Bergen
- Conferences
- Pair programming

Time

When do I have time to all this?

- Make time?
- Make sure you have “me time”

Experimentation

Think different

- Look for patterns
- Don’t stop experimenting
- Learn other languages
- Break old habits

Rust

Swift

Books

Ditch some old PHP books (except PHP 6 books)

- Domain-Driven Design in PHP - *Carlos Buenosvinos, Christian Soronellas and Keyvan Akbary*
- Dive Into Design Patterns - *Alexander Shvets*
- Front Line PHP - *Brent Roose*
- Thinking Functionally in PHP - *Larry Garfield (crell)*
- Object Design Style Guide - *Matthias Noback*
- The Clean Architecture in PHP - *Kristopher Wilson*

- Modernizing Legacy Applications In PHP - *Paul M. Jones*
- *Refactoring Improving the Design of Existing Code - Martin Fowler*
- *Clean Architecture - Robert C Martin (uncle Bob)*
- *Clean Code - Robert C Martin (uncle Bob)*
- *Domain Driven Design - Tackling Complexity in the Hart of Software - Eric Evans*
- *Advanced Web Application Architecture - Matthias Noback*

- Patterns Of Enterprise Application Architecture - *Martin Fowler*
- Design Patterns: Elements of Reusable Object-Oriented Software - *Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides*
- php[architect] (Magazine)

Get started with unit tests
Wed 28 Feb
https://www.meetup.com/php-bergen/events/298902700