From PHP to Jennifer



A guide for PHP developers: what carries over, what is different, and where the familiar tools went.

by @mplx

What feels familiar

- Variables start with `$`
- Braces, semicolons and `#` comments
- `"Hello, {$name}"` interpolates, `'...'` does not
- `try` / `catch` / `throw`, and a strict `match`
- Arrays are copied on assignment

Declare before you use

```
$count = 0;
$count = $count + 1;

def count as int init 0;
$count = $count + 1;
```

No type juggling

Expression	PHP	Jennifer
"10" == 10	true	false: a string is not an int
"5" + 1	6	error: + needs numbers
if ($count) with an int	truthy	error: a condition is a bool
(int) "42"	42	convert.toInt("42")

Strings

```
$greeting = "Hello, " . $name . "!";
echo "Hello, {$name}!\n";
```

```
use io;

def name as string init "Ada";
def greeting as string init "Hello, " + $name + "!";
io.printf("Hello, {$name}!\n");
```

One array becomes two types

```
use io;

def names as list of string init ["Ada", "Alan"];
$names[] = "Grace";

def prices as map of string to float init {"tea": 2.5, "cake": 3.9};
$prices["coffee"] = 3.2;
for (def item in $prices) {
    io.printf("%s|pad=8 %f|prec=2\n", $item, $prices[$item]);
}
```

Objects become values

```
use io;

def struct Account { owner as string, balance as int };

func deposit(a as Account, amount as int) {
    $a.balance = $a.balance + $amount;
    return $a;
}

def acc as Account init Account{owner: "Ada", balance: 100};
$acc = deposit($acc, 50);
io.printf("%s has %d\n", $acc.owner, $acc.balance);
```

Numbers keep their promises

Expression	PHP	Jennifer
7 / 2	3.5	3.5
intdiv(7, 2)	3	7 // 2 gives 3
-7 % 3	-1	2, floored like //
PHP_INT_MAX + 1	a float	error: integer overflow

match, as in PHP 8

```
use io;

def status as int init 404;
match ($status) {
    when 200 { io.printf("ok\n"); }
    when 404, 410 { io.printf("gone\n"); }
    else { io.printf("other\n"); }
}
```

Exceptions, without finally

```
use io;

func withdraw(balance as int, amount as int) {
    if ($amount > $balance) {
        throw Error{kind: "funds", message: "too little"};
    }
    return $balance - $amount;
}
try {
    withdraw(10, 50);
} catch (e) {
    io.printf("%s: %s\n", $e.kind, $e.message);
}
```

include, Composer and Packagist

PHP	Jennifer
require 'util.php';	import "./util.j" as util;
extension, ext-json	use json; (built in)
composer require x/y	jvc add "@scope/deck"
composer.lock	camcorder.lock
Packagist	registry.jennifer-lang.dev

The web: a server, not a request

```
import "web.j" as web;

func hello(ctx as web.Context) {
    def name as string init web.param($ctx, "name");
    web.text($ctx, 200, "Hello, {$name}!\n");
}

def app as web.App init web.new();
$app = web.get($app, "/hello/:name", hello);
web.run($app, "127.0.0.1:8080");
```

Databases without PDO

```
def db as sql.Connection init sql.open("postgres", $dsn);
def rows as sql.Rows init sql.query(
    $db, "select id, name from users where id > $1", 0);
while (sql.next($rows)) {
    io.printf("%d %s\n", sql.asInt($rows, "id"), sql.asString($rows,
"name"));
}
```

Where the tools went

PHP	Jennifer
php -l	jennifer lint
PHP-CS-Fixer	jennifer fmt -w
PHPUnit	jennifer test
Xdebug profiler	jennifer profile
php -a	jennifer repl

Cheat sheet

PHP	Jennifer
$x = 5;	def x as int init 5;
$x++;, $x += 1;	$x = $x + 1;
&&, ||, !	and, or, not
"a" . "b"	"a" + "b"
count($xs), strlen($s)	len($xs), len($s)
foreach ($m as $k => $v)	for (def k in $m) with $m[$k]
// comment	# comment

Where to go next

- The user guide: docs.jennifer-lang.dev
- The language tour: *Jennifer for coders*, in the Jennifer Academy
- Decks for APIs, file formats and more: registry.jennifer-lang.dev

Based on Jennifer 0.26.0, November 2026.