Hàm khởi tạo trong class

Đc chạy ngay khi 1 object mới đc tạo ra.

```
class A {

    function __construct() {

        echo 1;

    }
}
```

$nam = new A();

(!)

```
function __construct ($name, $age){

    $this -> name = $name;
    $this -> age = $age
}

$a = new A ('thienth", 30);

=) $a -> name -> 'thienth".
      -> age = 30
```

Hằng số trong class

```
$a = 5;
$a = 6;
```

```
const NAME = "thienth";

define("NAME", "thienth");

echo NAME;
```

H/số & thế gọi lại gtvi
class — mỗi class đc phép taoro h/số a phạm vi
tại dạng thuộc vì class.

```
class A {                        :: hás
    const NAME = "dkienth";         tính (static)
                    echo A :: NAME;
}

class B {
    const NAME = "namht";
                echo B :: NAME;

}
```

$B :: X() \to Y()$

gọi hàm vô hạn

$\to Z()$

$\to var();$ ⟹ static

Static $X()$
 return $model;

function $Y()$
 return $this

function $Z()$
 return $this

Base Model $\longrightarrow$ User

::where($1, $2, $3)

$x = `select * from`

Product.

Category.

users

$table = products

categories.

$model = table
where $1, $2 $3

function get()

connect | execute
prepare | fetchAll

Product::where(price, >, 3000)

$\Rightarrow$ select * from products
where price > 3000.

$prods = Product::where()->get()

+) Trừu tượng: cụ thể các mô tả về đối tượng trong g° cấu p/mềm → Class (tạo class)

+) Kế thừa: lớp B kế thừa (extends) lớp A. thì các Obj của lớp B đc phép sdụng thuộc tính/phg thức đc tạo ra từ lớp A.

+) Đa hình (overwrite): các lớp con đc phép đ/n lại gtị/nội dung của các ttính/phg thức đc đ/n ở lớp cha.

```php
class A {
    function run() {
        return "im runnig";
    }
}
```

```php
$x = new B()
$x -> name = "phat";
$x -> run()
=> phat running
```

```php
class B extends A {
    function run() {
        return $this -> name . "running";
    }
}
```

7) Đóng gói: phạm vi đặc phép tác động lên tính h/phụ thức trong class.

	everywhere	class con	nội đln (class)
public	✓	✓	✓
protected	✗	✓	✓
private	✗	✗	✓

```
class A {
    private $name = "phat gay";
}

$y = new A();

echo $y -> name => null


class B extends A {
    public function getName() {
        return $this->name;
    }
}

$x = new B()
$x->getName();
    => null
```

```
static function A(){
        $this x
        $model = new Static();

        $model -> get()
              -> run()
              -> eat().
}
```