

`statement->fetch()` “Ne prend qu’une seule ligne à la fois il faut boucler pour prendre la suivante et ainsi de suite.”

`statement->fetchAll()` “Prend toutes les lignes en une seule fois”.

Par défaut si on ne précise pas au fetch() si l'on veut récupérer les colonnes par index c'est à dire colonne[0], colonne[1]..etc ou par nom de colonne (tableau associatif) c'est à dire first\_name, last\_name etc..  
Il nous retourne les deux,

```
index.php x
1  <?php
2
3
4  try{
5      $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
6      $dataBase->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
7  }catch(Exception $e){
8      die('ERROR: '.$e->getMessage());
9  }
10
11
12  $results = $dataBase->query( statement: 'SELECT first_name, last_name FROM customers');
13
14  var_dump($results->fetch());
15
```

par nom.

et

par index.

localhost/projetStore/php/

```
array(4) { ["first_name"]=> string(8) "Michelle" [0]=> string(8) "Michelle" ["last_name"]=> string(5) "Smith" [1]=> string(5) "Smith" }
```

Expand All Collapse All

```
(array) [4 elements]
  first_name: (string) Michelle
  0: (string) Michelle
  last_name: (string) Smith
  1: (string) Smith
```

Si l'on veut chercher nos  
information par nom de colonne  
uniquement .

```
index.php x
1 <?php
2
3
4 try{
5     $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
6     $dataBase->setAttribute( attribute: PDO::ATTR_ERRMODE , value: PDO::ERRMODE_EXCEPTION);
7 }catch(Exception $e){
8     die('ERROR: '.$e->getMessage());
9 }
10
11
12 $results = $dataBase->query( statement: 'SELECT first_name, last_name FROM customers');
13
14 var_dump($results->fetch( fetch_style: PDO::FETCH_ASSOC));
15
16
17 /*while($row = $results->fetch()){ // tant que result peut faire la methode fetch
18     echo $row['first_name'].'<br>'; //sur de la donné et valoriser $row.
19 }*/
20
21 ?>
```

```
array(2) { ["first_name"]=> string(8) "Michelle" ["last_name"]=> string(5) "Smith" }
```

▼Expand All ▲Collapse All

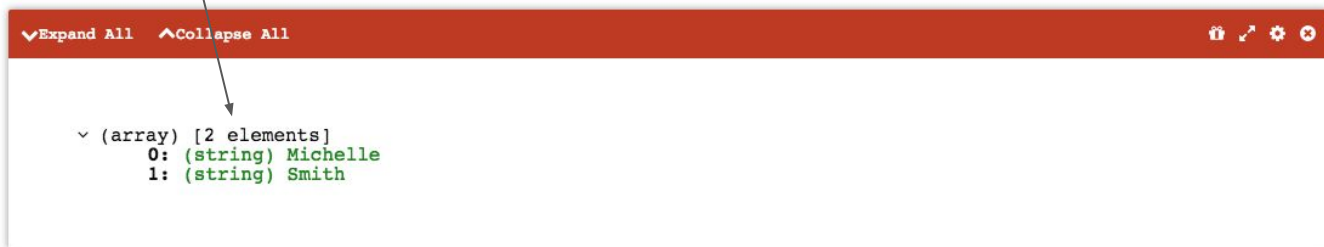
```
▼ (array) [2 elements]
    first_name: (string) Michelle
    last_name: (string) Smith
```

Si l'on veut chercher nos information par numéro de colonne uniquement .

```
index.php x
1 <?php
2
3
4 try{
5     $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
6     $dataBase->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
7 }catch(Exception $e){
8     die('ERROR: '.$e->getMessage());
9 }
10
11
12 $results = $dataBase->query( statement: 'SELECT first_name, last_name FROM customers');
13
14 var_dump($results->fetch( fetch_style: PDO::FETCH_NUM));
15
16
17 /*while($row = $results->fetch()){ // tant que result peut faire la methode fetch
18     echo $row['first_name'].'<br>'; //sur de la donnée et valoriser $row.
19 }*/
20
21 ?>
```

localhost/projetStore/php/

```
array(2) { [0]=> string(8) "Michelle" [1]=> string(5) "Smith" }
```



On doit appliquer la méthode ->fetch() pour aller chercher la data de notre statement \$results ce que l'on récupère est toujours un tableau donc **il faut l'appeler en tant que tel c'est à dire avec une boucle.**

```
index.php x
1 <?php
2
3
4 try{
5     $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
6     $dataBase->setAttribute( attribute: PDO::ATTR_ERRMODE , value: PDO::ERRMODE_EXCEPTION);
7 }catch(Exception $e){
8     die('ERROR: '.$e->getMessage());
9 }
10
11
12 $results = $dataBase->query( statement: 'SELECT first_name, last_name FROM customers');
13
14 $variable = $results->fetch();
15
16
17 $row = $results->fetch();
18
19 while($row){ // tant que fetch() me retourne une valeur row vaut true sinon il vaut false
20     echo $row['first_name']." ".$row['last_name'].'<br>';
21     $row = $results->fetch();
22 }
23
24
```

← → ↺ ⓘ local

Ben  
William  
Sarah  
Donald  
Beth  
Henry  
Corin  
George  
Karen  
Warren  
Steven  
Bob  
Shawn  
Mark  
Barry  
Roland  
Christian  
Kelly  
Greg  
Brad  
Peter  
Steven  
Mathieu

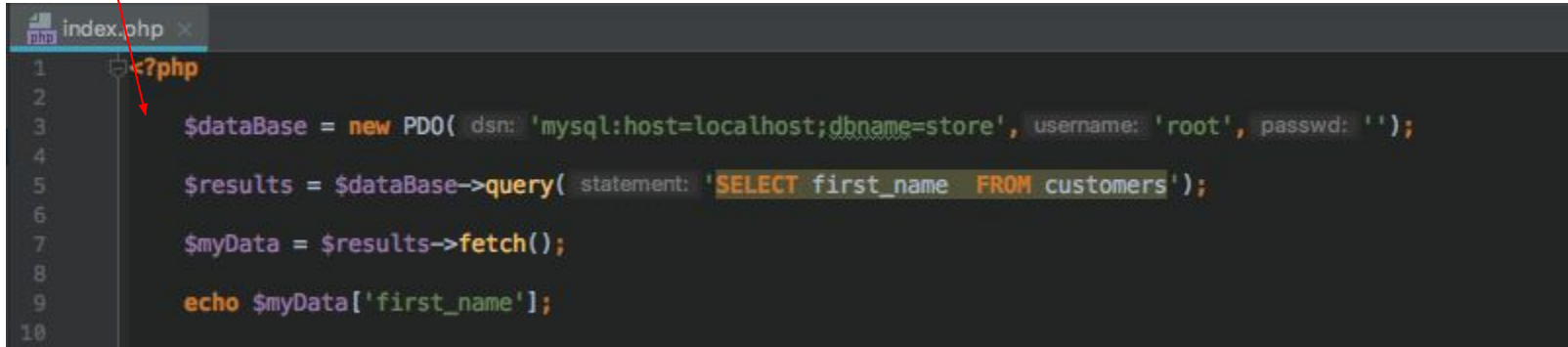
## Syntaxe différente mais même boucle

```
index.php x
1 <?php
2
3
4 try{
5     $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
6     $dataBase->setAttribute( attribute: PDO::ATTR_ERRMODE , value: PDO::ERRMODE_EXCEPTION);
7 }catch(Exception $e){
8     die('ERROR: '.$e->getMessage());
9 }
10
11
12 $results = $dataBase->query( statement: 'SELECT first_name, last_name FROM customers');
13
14 $variable = $results->fetch();
15
16
17 while($row = $results->fetch()){ // tant que fetch() me retourne une valeur tu continue à boucler
18     echo $row['first_name'].'<br>';
19 }
20
21
```

← → ↺ ⓘ local

Ben  
William  
Sarah  
Donald  
Beth  
Henry  
Corin  
George  
Karen  
Warren  
Steven  
Bob  
Shawn  
Mark  
Barry  
Roland  
Christian  
Kelly  
Greg  
Brad  
Peter  
Steven  
Mathieu

exemple d'appel que d'une seul ligne car fetch() ne récupère qu'une ligne si on ne boucle pas.



A screenshot of a code editor window titled 'index.php'. The code is as follows:

```
1 <?php
2
3 $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
4
5 $results = $dataBase->query( statement: 'SELECT first_name FROM customers');
6
7 $myData = $results->fetch();
8
9 echo $myData['first_name'];
10
```

A red arrow points from the text above to the `fetch()` method call on line 7.

statement->fetchAll() nous permet de chercher toute les données de notre requête(statement) en une fois.

```

1  <?php
2
3
4  try{
5      $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
6      $dataBase->setAttribute( attribute: PDO::ATTR_ERRMODE , value: PDO::ERRMODE_EXCEPTION);
7  }catch(Exception $e){
8      die('ERROR: '.$e->getMessage());
9  }
10
11
12  $results = $dataBase->query( statement: 'SELECT first_name, last_name FROM customers');
13
14  $variable = $results->fetchAll();
15
16  var_dump($variable);
17
18

```

```
ray(24) { [0]=> array(4) { ["first_name"]=> string(8) "Michelle" [0]=> string(8) "Michelle" ["last_name"]=>  
=> string(3) "Ben" ["last_name"]=> string(5) "Barro" [1]=> string(5) "Barro" } [2]=> array(4) { ["first_nam  
ing(5) "Terry" } [3]=> array(4) { ["first_name"]=> string(5) "Sarah" [0]=> string(5) "Sarah" ["last_name"]=>  
onald" [0]=> string(6) "Donald" [1]=> string(4) "Barro" [2]=> string(4) "Barro" [3]=> string(4) "Grog"  
]=> string(7) "Mc  
ing(5) "Corin" [0]  
ing(6) "Harris" [1  
first_name"]=> str  
last_name"]=> str  
acramento" } [13  
dark" [0]=> string  
]=> string(5) "Wh  
ing(9) "Christian"  
last_name"]=> str  
0]=> array(4) { ["  
ing(5) "Peter" [1  
ing(8) "Thompso
```

▼ Expand All ▲ Collapse All

```
▼ (array) [24 elements]  
  ▼ 0: (array) [4 elements]  
    first_name: (string) Michelle  
    0: (string) Michelle  
    last_name: (string) Smith  
    1: (string) Smith  
  ▼ 1: (array) [4 elements]  
    first_name: (string) Ben  
    0: (string) Ben  
    last_name: (string) Barro  
    1: (string) Barro  
  ▼ 2: (array) [4 elements]  
    first_name: (string) William  
    0: (string) William  
    last_name: (string) Terry  
    1: (string) Terry  
  ▼ 3: (array) [4 elements]  
    first_name: (string) Sarah  
    0: (string) Sarah  
    last_name: (string) Thompson  
    1: (string) Thompson  
  ▼ 4: (array) [4 elements]  
    first_name: (string) Donald  
    0: (string) Donald  
    last_name: (string) Duck  
    1: (string) Duck
```



```

first_name"]=>str
["last_name"]=>str
"Sacramento" ]|3
"Mark" [0]=> string
1]=> string(5) "Wh
string(9) "Christia
["last_name"]=> str
[20]=> array(4) [ ]
string(5) "Peter" [l
string(8) "Thompso
Ben Barro
William Terry
Sarah Thompson
Donald Duck
Beth McAdams
Henry Wallace
Corin Sabbath
George Harris
Karen White
Warren Bartlet
Steven Wright
Bob Sacramento
Shawn Hurley
Mark Roast
Barry White
Roland Groburg
Christian Hayhouse
Kelly Bark
Greg Thompson
Brad Traversy
Peter Smith

```

Note : `fetchAll()` nous retourne un tableau, ce tableau comporte des lignes (qui sont en fait des tableaux) qui ont forcément des colonnes, `index [0] = "mathieu"` ou `first_name => "mathieu"` etc..

## Exemple INSERT INTO

```
index.php x
1  <?php
2
3
4  try{
5      $dataBase = new PDO( dsn: 'mysql:host=localhost;dbname=store', username: 'root', passwd: '');
6      $dataBase->setAttribute( attribute: PDO::ATTR_ERRMODE , value: PDO::ERRMODE_EXCEPTION);
7  }catch(Exception $e){
8      die('ERROR: '.$e->getMessage());
9  }
10
11
12  $dataBase->query( statement: "INSERT INTO customers(first_name, last_name) VALUES ('Lea', 'Kent')");
13
14
15
16
17  ?>
```