

# モバイル プログラミング2

# 教材をDLしてください

## <https://git.io/vMZ5H>

※以下のURLへリダイレクトします。

<https://github.com/katsube/neec/blob/master/mobileprogramming2/20170116/api.zip>

本日の予定

# 午前

- 前年の質問に答えるコーナー
- RESTful API
  - 通信内容をのぞいて見る
  - JSON
  - APIを作成する

# 午後

- 試験の説明
  - 範囲、内容
- オンライン対戦ゲーム
  - 開発内容の説明
  - 進め方の説明
  - 製作開始

前回休んだ人

(°▽°)ｼ

PC借りた人  
(°▽°)ｼ

# まずは追いつこう

1. GitHubの資料見てね

<http://github.com/katsube/neec>

2. 環境構築

3. 環境構築で困ったらすぐに聞いてください



# アンケート (出席カード)

# アンケート

1. 提出 = 出席 (授業終了までに限る)  
**未提出 = 欠席**
2. 学籍番号、名前が確認できない場合は**欠席**
3. わからない場合は、どこが理解できなかったか記入

# アンケート

1. 「白紙提出」「授業を聞いていたと判断できない」場合は個別にヒアリングを行います。

- よほどのことがなければ呼び出されません
- 大人としての自覚を持って授業に望んで下さい。

2. 一人では解決できないことがある場合、自分から聞きにくるよう。

# アンケート

- 返却を希望する場合

モバイルプログラミング 2 出席カード

2016/10/31

学籍番号 [                      ]      氏名 [                      ]

☐ 返却を希望する

問題 1. PHP の特徴を各項目毎にまとめてみましょう

実行方法 ※大きく 2 種類

次回～次々回の授業で返却します

# アンケート

- 難易度に○をつける

モバイルプログラミング 2 出席カード

2016/11/07

学籍番号 [                      ]      氏名 [                      ]

難易度 ( 優しい ・ 最適 ・ 難しい )

☐ 返却を希望する

○をつけてください。  
様子を見て難易度を調整します。

アンケートに  
答えるコーナー

# 質問

1. (SQLの)結合が理解できなかったので問題集などがあれば教えてほしい
2. プルリクについて知りたい

# RESTful API

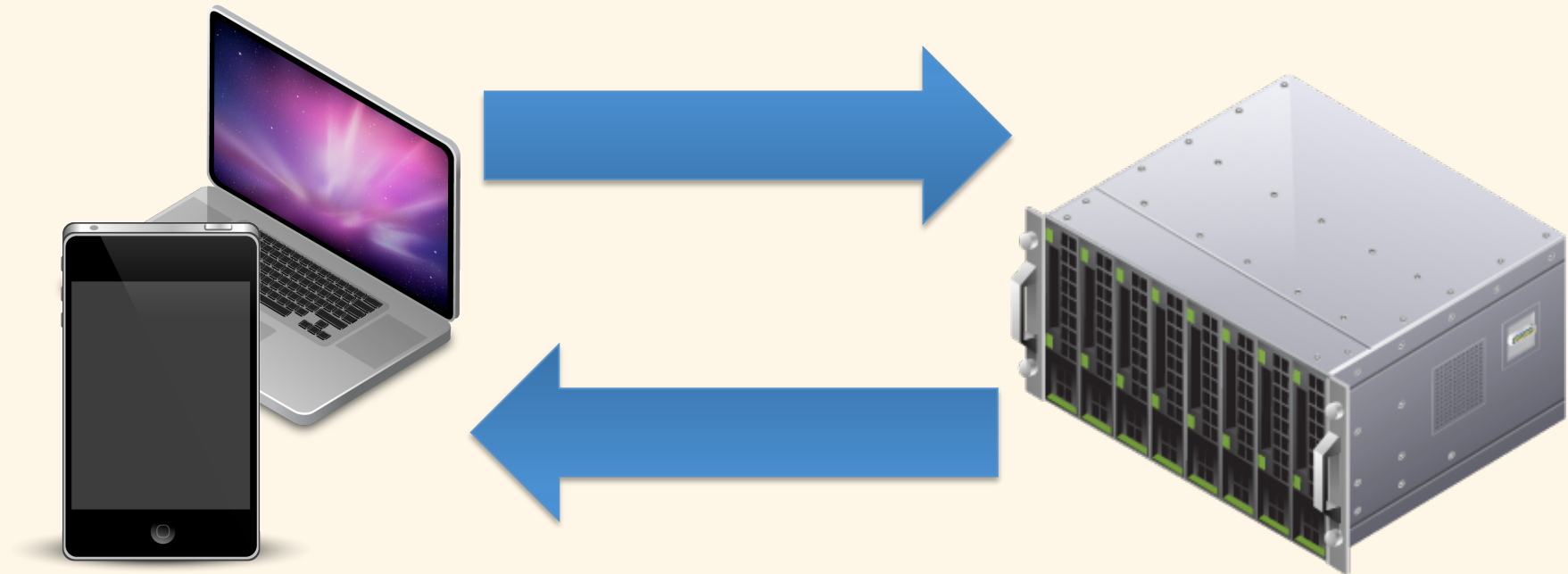




# クライアント・サーバ間通信

クライアント

サーバ



※昨今ではHTTP(S)の上で通信を行うことが多い

# 実際に通信内容をのぞいて見る

SQUARE ENIX.

コンテンツ一覧 +

YAHOO! JAPAN ゲーム

本ゲームは Yahoo! ゲームで展開しています。

プレイフリーデフォルト

## BRAVELY DEFAULT

PRAYING BRAGE

— あの世界から200年 —  
ブラウザゲームで  
新章開幕



WebMoney Award 2013  
Rookie of the Year  
BEST GAMES

<http://yahoo.bravely.jp/>

START  
Yahoo! JAPAN IDを  
お持ちでない方はこちら

BRAVELY DEFAULT とは

FAQ

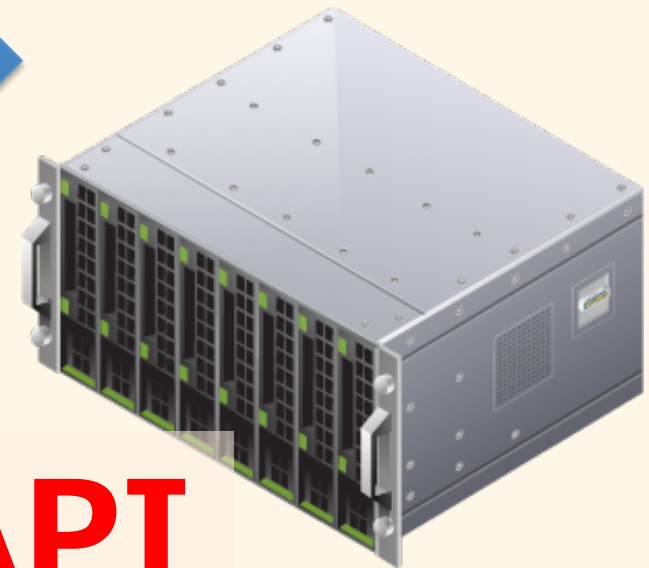
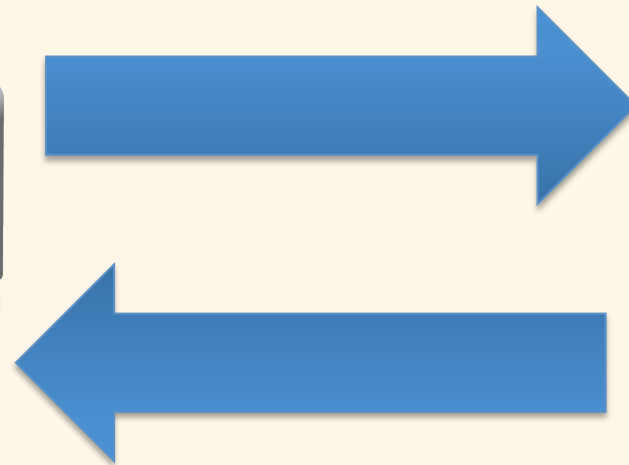
利用規約

プライバシーポリシー

# クライアント・サーバ間通信

クライアント

サーバ



**RESTful API  
(REST API)**

# クライアント・サーバ間通信

クライアント

サーバ

- ・ URL (EndPoint)
- ・ その都度
- ・ JSON形式



※JSON形式が最近は多いですが、XMLや他のデータフォーマットの場合もあります。



# JSON

## JavaScript Object Notation

JavaScriptの定数、変数、配列等の記述方法を元にしたデータフォーマットの一種。昨今ではデファクトと呼べるほど広く普及している。

JavaScript以外にも多数の言語にライブラリが用意されており、大抵の場合かんたんに扱うことができる。

<http://www.json.org/>

<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>

# JavaScriptの記述方法のおさらい

- **数字**  
1, 2, 3 ...
- **文字列**  
"Hello"  
'World'
- **真偽値**  
true, false
- **null**  
null
- **配列**  
[1, 2, 3, "Foobar"]
- **連想配列**  
{"foo":1, "bar":2}
- **関数**  
function foo(arg){  
 ...  
}  
function (arg){  
 ...  
}

# JSON例

```
{id:1, name:"ババ\ムート"}
```

# JSON例

```
[  
  {id:1, name:"バウムート"},  
  , {id:2, name:"チョコボ"},  
  , {id:1, name:"バウムート"},  
]
```



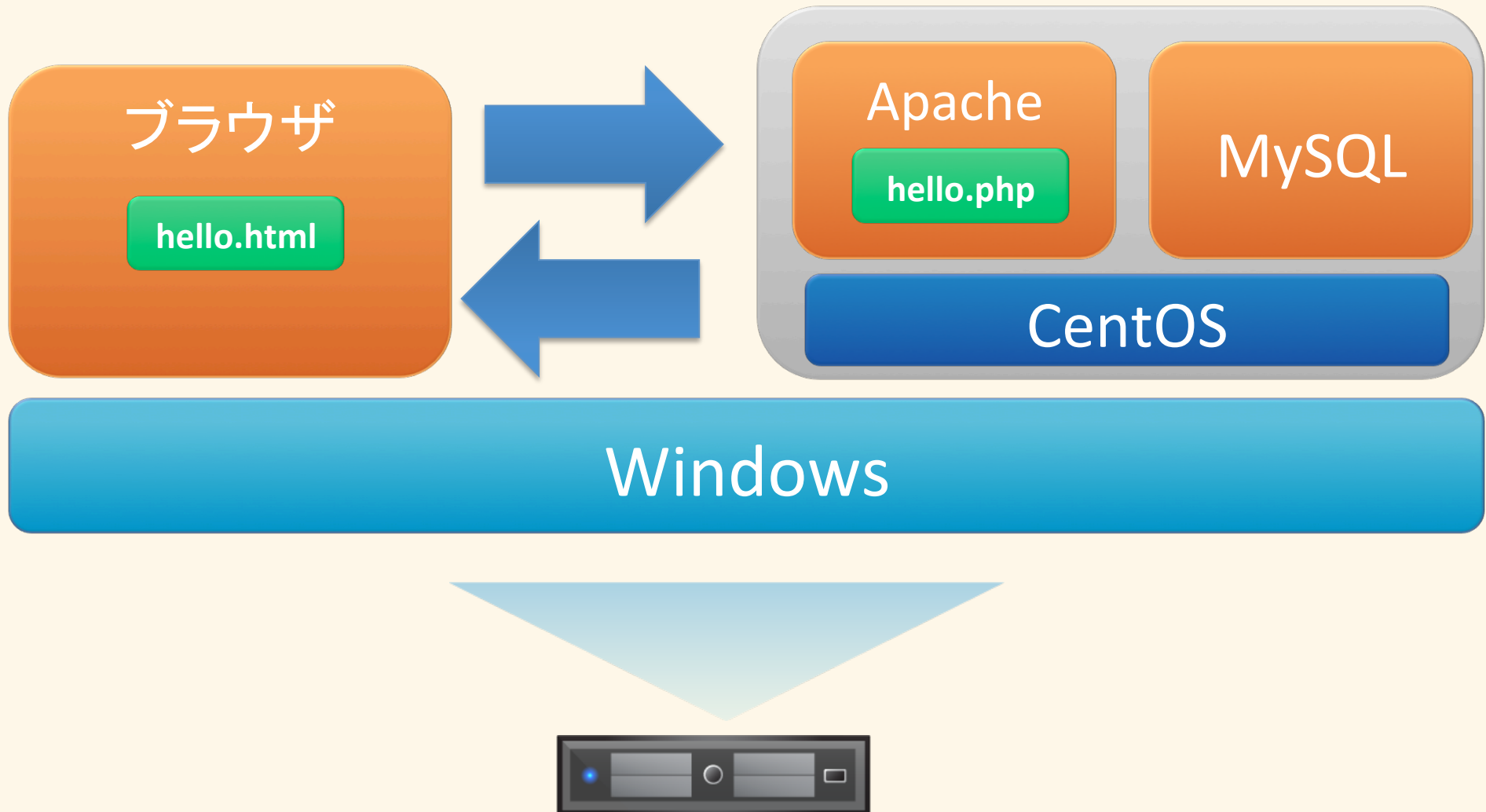
# JSON例

```
{  
  head:{  
    status: "success"  
  },  
  gacha:[  
    {id:1, name:"バラムート"},  
    , {id:2, name:"チョコボ"},  
    , {id:1, name:"バラムート"},  
  ]  
}
```

# 実際にAPIを試してみる

クライアント

サーバ



# 教材をDLしてください

## <https://git.io/vMZ5H>

※以下のURLへリダイレクトします。


<https://github.com/katsube/neec/blob/master/mobileprogramming2/20170116/api.zip>

# サーバのIPアドレスを確認

\$ ifconfig

```
[neec@localhost api]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe90:c13c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:90:c1:3c txqueuelen 1000 (Ethernet)
    RX packets 142265 bytes 144265285 (137.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 45802 bytes 3015869 (2.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.60.235.27 netmask 255.224.0 broadcast 10.60.255.255
    inet6 fe80::a00:27ff:fe90:c13c prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:4d:f1:10 txqueuelen 1000 (Ethernet)
    RX packets 42388 bytes 4724173 (4.5 MiB)
    RX errors 0 dropped 17 overruns 0 frame 0
    TX packets 3432 bytes 413759 (404.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



# APIを作ってみる (Server)

/var/www/html/**api/hello.php**

```
<?php
$data = [
    'str1'=>'Hello'
    , 'str2'=>'World'
];
$json = json_encode($data);

print $json;
```

# APIを作ってみる (Client)

## Windowsの任意の場所

```
<div id="result"></div>

<script src="js/jquery-3.1.1.min.js"></script>
<script>
$(document).ready(function(){
    $.ajax({
        url: "http:// 【IPアドレス】 /api/hello.php"
        , dataType: "json"
        , success: function( json ) {
            $("#result").html( json["str1"] + json["str2"] + "!" );
        }
    });
});
</script>
```

# APIを作ってみる (Client)

## Windowsの任意の場所

```
<div id="result"></div> ←結果を格納するタグ
```

```
<script src="js/jquery-3.1.1.min.js"></script> ←外部のJSを読み込む
```

```
<script>
```

```
$(document).ready(function(){ ←読み込み完了のイベントで実行
```

```
    $.ajax({
```

```
        url: "/api/hello.php" ←APIのURL
```

```
        , dataType: "json" ←APIが返却するデータ形式
```

```
        , success: function( json ) { ←APIへのリクエストが成功時successを実行
```

```
            $("#result").html( json["str1"] + json["str2"] + "!" );
```

```
        }
```

```
    });
```

```
});
```

```
</script>
```

# PHPとJSの比較 1



```
$array = [1,2,3];  
$hash = [  
    'str1'=>'Hello'  
    , 'str2'=>'World'  
];
```

```
$array[0];  
$hash['str1'];
```

```
array = [1,2,3];  
hash = {  
    'str1':'Hello'  
    , 'str2':'World'  
};
```

```
array[0];  
hash['str1'];
```



# PHPとJSの比較 2



```
$mix = [  
    'str'=>'Hello'  
    , 'f' => function($a){  
        print $a;  
    }  
];
```

```
$mix['f']('foo');
```

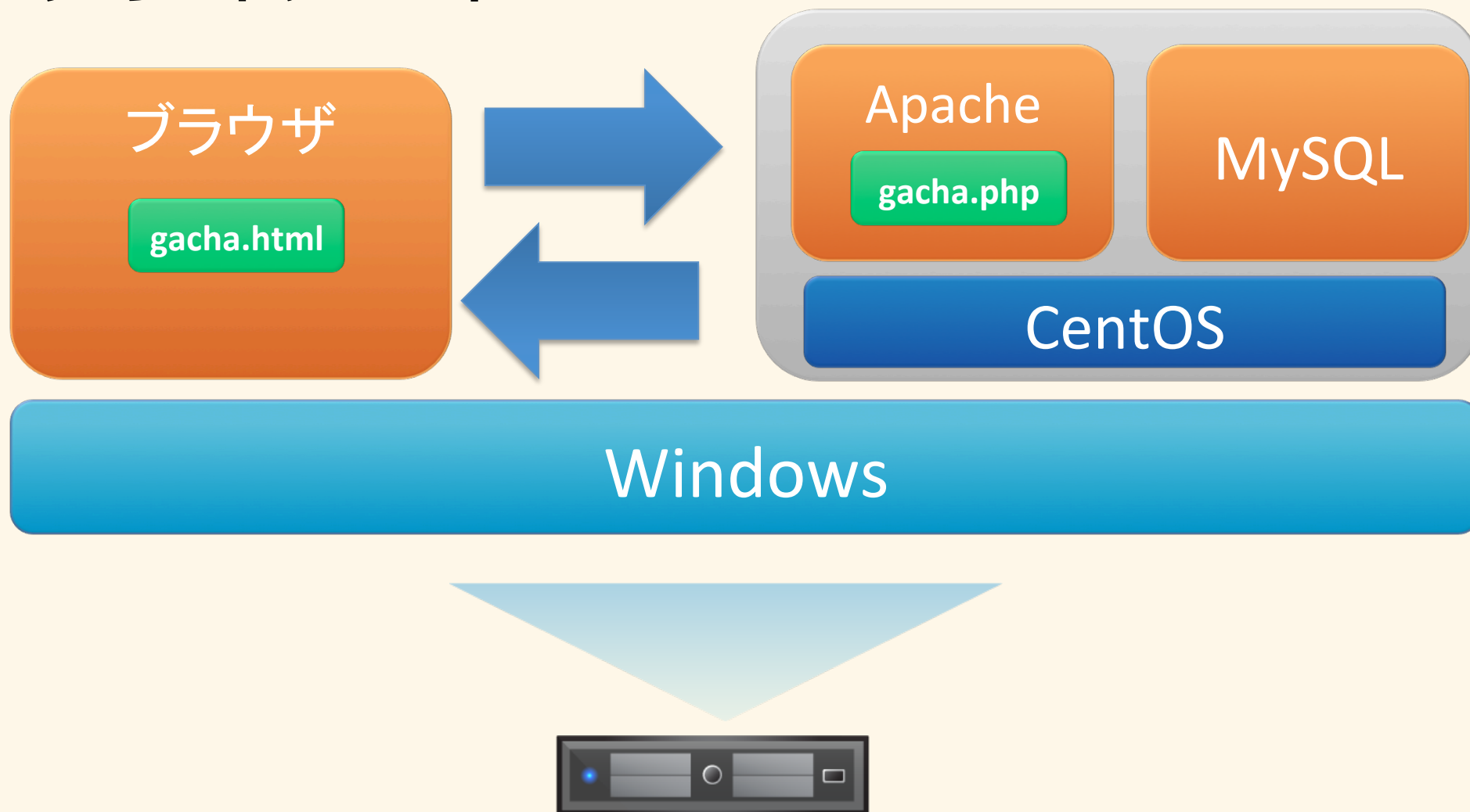
```
mix = {  
    'str':'Hello'  
    , 'f': function(a){  
        alert(a);  
    }  
};
```

```
mix['f']('foo');  
mix.f('foo');
```

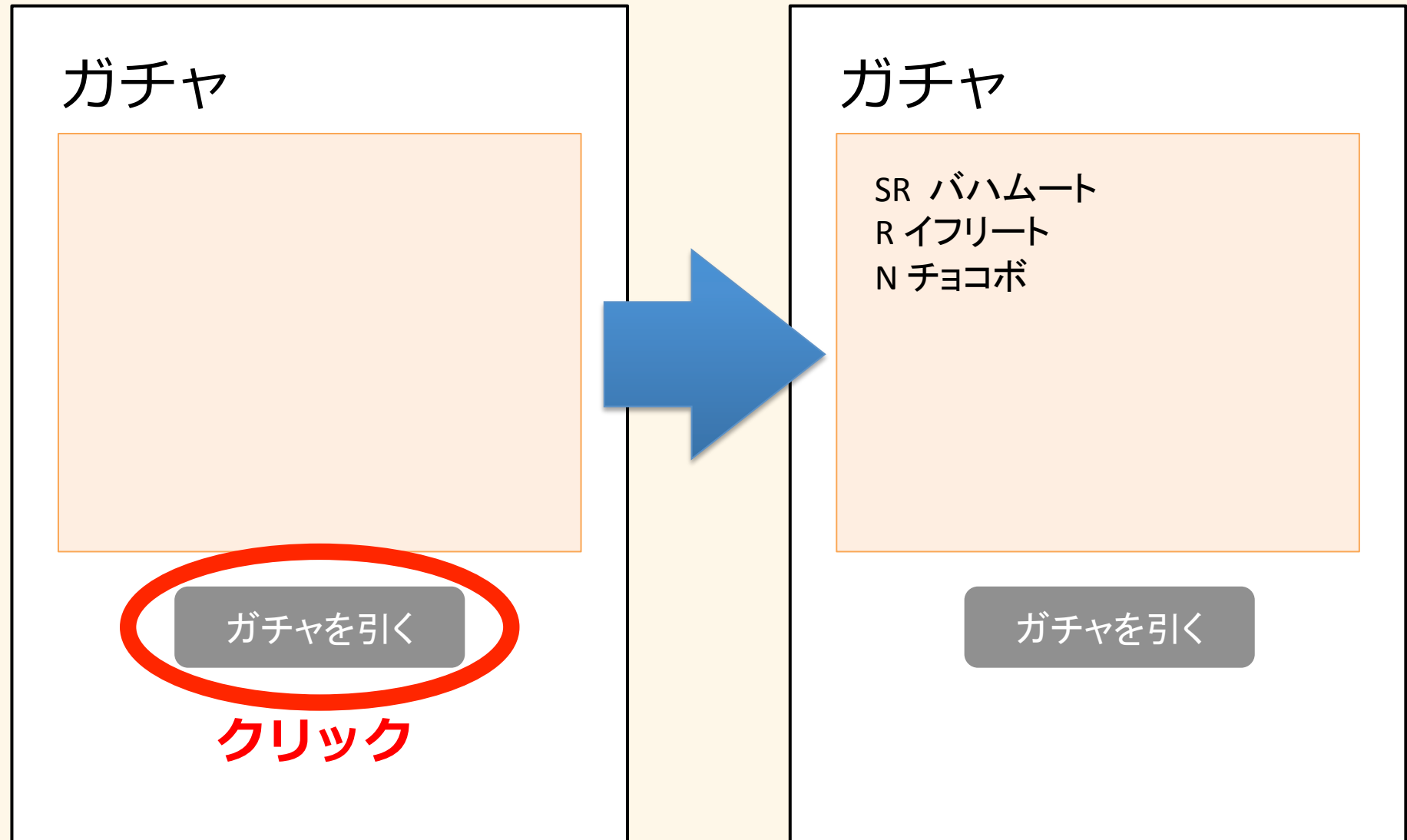
# ガチャAPIを作成する

クライアント

サーバ



# 演習. ガチャAPI 1



# 演習. ガチャAPI 2

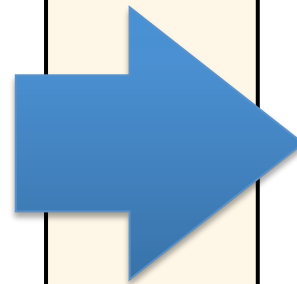
ガチャ



5 回

ガチャを引く

クリック



ガチャ

SR バハムート  
R イフリート  
N チョコボ  
R イフリート  
N チョコボ

5 回

ガチャを引く

# 演習. ガチャAPI 3

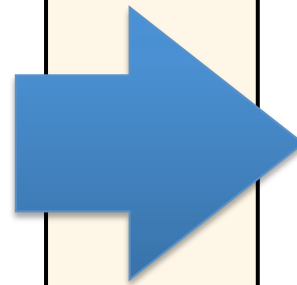
ガチャ

所持カード

クリック

5 回

ガチャを引く



ガチャ

所持カード

SR バハムート 1枚  
R イフリート 2枚  
N チョコボ 3枚

5 回

ガチャを引く