

```
1. // 再帰的な呼び出し（ブラウザにpタグを連続で追加）
2.
3. // -----
4. function pPut1() {
5.     $("body").append("A");
6.     pPut1();
7. }
8. // pPut1();
9. //無限ループ発生、ブラウザクラッシュ
10.
11.
12. // -----
13. function pPut2(i) {
14.     if (i < 10) {
15.         $("body").append("B");
16.         pPut2(i + 1);
17.     }
18. }
19. // pPut2(0);
20. //繰り返しをif文と引数で制御、同時に10個出現
21.
22.
23. // -----
24. function pPut3(i) {
25.     if (i < 10) {
26.         setTimeout(function() {
27.             $("body").append("C");
28.             pPut3(i + 1);
29.         }, 100);
30.     }
31. }
32. // pPut3(0);
33. //setTimeoutで0.1秒のウェイトをつけて再帰的に呼び出す
```

```

1. // ランダムおみくじアプリの作成
2.
3. // [1]-----
4. let color = ["赤", "青", "黄"];
5. $("button").on("click", function() {
6.   $("div").text(color[0]);
7. });
8.
9. // let : 同じブロック内で再定義不可な変数宣言
10. // 配列の基本
11. // buttonにclickイベントリスナーを登録する
12. // textメソッドで要素内の文字を変更する
13.
14.
15.
16. // [2]-----
17. $("button").on("click", function() {
18.   let r = Math.floor(Math.random() * 3);
19.   $("div").text(color[r]);
20. });
21.
22. // Math関数による、ランダムな整数のとりだし
23.
24.
25.
26. // [3]-----
27. function roll(n) {
28.   setTimeout(() => {
29.     if (n < 50) {
30.       let r = Math.floor(Math.random() * 3);
31.       $("div").text(color[r]);
32.       roll(n + 1);
33.     }
34.   }, 100);
35. }
36. $("button").on("click", function() {
37.   roll(0);
38. });
39.
40. // 再帰的な関数実行を組み込んで、ルーレットにする
41. // 引数nが50になるまで、0.1秒ごとに繰り返す
42.

```

```
1. //同期通信による時計アプリ
2.
3. ----[index.php]-----
4. <div class="clock">
5.   <?php require("watch.php"); ?>
6.   <!--requireでバックエンドからファイルを呼び出す-->
7. </div>
8.
9. <!--jQuery-->
10. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
11.
12. <script>
13.   setTimeout(function() {
14.     window.location.reload();
15.   }, 1000);
16. // ページを1秒ごとにリロードする
17. </script>
18. -----
19.
20.
21. [watch.php]-----
22. <?php
23.   date_default_timezone_set("asia/Tokyo");
24.   echo date("Y年m月d日 H:i:s");
25. ?>
26. -----
27.
28. <!--これだと、ブラウザ全体が更新されるのでユーザーはほかの操作ができない-->
```

```

1. //非同期通信による時計アプリ
2.
3. ----[index.php]-----
4. <div class="clock">
5. </div>
6.
7. <!--jQuery-->
8. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
9.
10. <script>
11.     function updateTime() { //updateTime関数を定義します
12.         $.get("watch.php", function(data) {
13.             //$.get("外部URL",コールバック関数);
14.             // dataにはwatch.phpでechoされたものが入る
15.             $(".clock").text(data);
16.             // .clock内に時刻を表示
17.             updateTime();
18.             // その後、自分自身（この関数）を再び呼び出す・再帰的な実行
19.         })
20.     }
21.
22.     setTimeout(updateTime, 1000);
23.     //1秒間隔でupdateTimeを実行
24. </script>
25. ----
26.
27.
28. [watch.php]-----
29. <?php
30.     date_default_timezone_set("asia/Tokyo");
31.     echo date("Y年m月d日 H:i:s");
32. ?>
33. ----
34.
35. <!--これならば、ブラウザの一部だけが更新されるのでユーザーはほかの操作ができる-->

```

1. *//phpをフォームから起動する*
- 2.
3. [index.html]-----
4. <!--テキストエリアと書き込みボタンフォームを作る-->
5. <form action="backend.php" method="POST">
6.   <textarea name="comment"></textarea>
7.   <input type="submit" value="書き込む">
8. </form>
9. -----
- 10.
11. [backend.php]-----
12. <?php
13.   echo \$\_POST["comment"];
14. ?>
15. -----
16. *// フォームを使ってのPOST送信の基本中の基本*
17. *//しかしながらこれだと、ブラウザが変遷してしまいます。*
18. *//次はブラウザを変遷せずに、テキストエリアの下にコメントを出そうと思います。*

1. //バックエンドのphpをフォームから起動し非同期通信でフロントエンドに返す

2.

3. [index.html]-----

4. <!--テキストエリアと書き込みボタンフォームを作る-->

5. <form action="backend.php" method="POST">

6.   <textarea name="comment"></textarea>

7.   <input type="submit" value="書き込む">

8. </form>

9. <div class="box"></div>

10. <!--jQuery-->

11. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

12. <script>

13.   \$("form").on("submit", function(e) {

14.     e.preventDefault();

15.     let formData = new FormData(this);

16.     \$.ajax({

17.         url: "backend.php",

18.         type: "POST",

19.         data: formData,

20.         contentType: false,

21.         processData: false,

22.         success: function(result) {

23.             \$(".box").append("<p>" + result + "</p>");

24.         }

25.     })

26.     })

27. </script>

28.

29. □function(e)のeはイベントオブジェクトといい、ユーザーイベントがあった時に自動的にeの中に色々な情報がセットされる

30. □e.preventDefault();は本来の挙動をキャンセルする（ここではフォーム送信でのブラウザ更新をキャンセル）

31. □let formData=new FormData(this);はフォームのコントロール（inputやtextareaのデータ）を格納する命令。

32. 難しく言うと、this（フォーム自体）のFormDataクラスのインスタンスとしてformDataを作ったという言い方をする。

33. □\$.ajaxは\$.getメソッドの進化系、()の中に

34. {

35.   キー:"値",

36.   キー:"値",

37. }

38. の形でプロパティを追加していく。

39. 主なキーは

40. type:"送信タイプ（POSTやGET）",

41. url:"送信先のファイルパス（URL）",

42. data:"送信するデータ", ...ここではformのinputやtextareから受け取ったデータである変数formData

43. success:function(data){

44.   //やりたい処理・dataにはバックエンドでechoされたものが入っている

45. }

```
46. -----
47.
48.
49. [backend.php]-----
50. <?php
51. echo $_POST["comment"];
52. ?>
53. -----
54. //$.ajaxで非同期にバックエンドからの情報を引き出しました。
55. //これでブラウザが変遷することなくフロントエンド側にデータが送られます。
56. //次はコメントを保存する機能をバックエンドでテキストファイルを作成して呼び出すことで実装しようと思いま
    す。
```

```

1. //メモ帳アプリ.ver1
2.
3. ---[index.html]-----
4. <form action="backend.php" method="POST">
5.   <textarea name="comment"></textarea>
6.   <input type="submit" value="書き込む">
7. </form>
8. <div class="box"></div>
9. <!--jQuery-->
10. <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
11. <script>
12.   // フォームを送信するときのコードです
13.   $("form").on("submit", function(e) {
14.     e.preventDefault();
15.     let formData = new FormData(this);
16.     $.ajax({
17.       url: "backend.php",
18.       type: "POST",
19.       data: formData,
20.       contentType: false,
21.       processData: false,
22.       success: function(result) {
23.         $(".box").empty().append("<p>" + result + "</p>");
24.         //一度表示エリアを空にしてから、受け取ったデータを表示する。
25.         $("form")[0].reset();
26.         // 書き込まれたフォームの内容をリセットする。
27.         // (reset();はjsバニラの構文なのでjQueryオブジェクトをjavascriptの純粋DOMに変換する $("X")[0])
28.       }
29.     })
30.   });
31.   // ブラウザを起動した瞬間のコード
32.   // $.getメソッドを使って受け取る
33.   $(document).ready(function() {
34.     $.get("backend.php", function(result) {
35.       $(".box").append("<p>" + result + "</p>");
36.     })
37.   })
38. </script>
39. -----
40.
41.
42. ----[backend.php]-----
43. <?php
44. if (isset($_POST["comment"])) :
45.   //サーバーからname="comment"のテキストエリア内容が送信 (POST) されていれば…
46.
47.   $comment = "<div>" . $_POST["comment"] . "</div>";
48.   //受け取ったコメントをdivタグで囲んだデータとして$commentに格納する

```



```

49.
50. //おまけ:改行および、空白も認識させたい場合
51. // $comment = "<div>". nl2br(str_replace(" ", " ", $_POST["comment"])). "</div>";
52.
53. date_default_timezone_set("asia/Tokyo");
54. $timestamp = date("Y年m月d日H時i分s秒");
55. //日本時刻のタイムスタンプを作成
56.
57. $fileName = "memo-{$timestamp}.txt";
58. // ファイルの名前を作成
59. // 文字列中の変数展開は""の中で{$変数名}とすると行える
60.
61. file_put_contents("data/$fileName", "$comment", FILE_APPEND);
62. // dataフォルダ内にファイルを書き込み
63. // FILE_APPENDオプションは今回はいらないが、一応。
64. // 既にあるファイル名を指定した場合にファイルの最後尾に追加で書き込みをする
65. endif;
66.
67. $fileNames = glob("data/*.txt");
68. // dataフォルダ内の.txtファイルすべての名前を配列で取得
69.
70. foreach ($fileNames as $fileName) {
71.     echo file_get_contents($fileName);
72. }
73. // ファイルがある分だけ、各ファイル名に対応するファイルの中身を取り出して出力する。
74. -----
75. //上記のプログラムをPHPが動くサーバーに置けば、簡易メモアプリが動きます。
76. //少し改変すれば画像や動画ファイルのアップローダーも作れます。
77. //次は任意のコメントを削除するコードを考えてみるといいでしょう。
78. //あとはcssでスタイルを整えれば完成です。

```