Web Design and Development
PHP 2

Name
Muhammad Baihaqi Aulia Asy'ari

NIM
2241720145

Class
2I

Department
Information Technology

Study Program
D4 Informatics Engineering

1. The text in the function printed in the page. Text was defined in the function and the function was called after it was defined.
2. The function was called twice and each call has their own name and greetings. The first call was directly written in the parameters. The second call was represented by a stored variables.
3. The default parameter will be used if the parameter with default value are not inlcuded in the method call.
4. When the function is called it subtitute it self with the return value.
5. A function can call other function and get its return value if available.
6. It print "Halo dunia!" on the page and doesnt stop printing.
7. It loops just like a normal for loop.
8. It print every item with the key "nama" as an item of unordered list.
9. -

```php
<?php
    // Experiment 9 - recursive tiered menu case study code
    ↪  challenge
    $menu = [
        [
            "name" => "Dashboard"
        ],
        [
            "name" => "News",
            "subMenu" => [
                [
                    "name" => "Touring",
                    "subMenu" => [
                        [
                            "name" => "Beach"
                        ],
                        [
                            "name" => "Mountain"
                        ]
                    ]
                ],
                [
                    "name" => "Culinary"
```

```
                ],
                [
                    "name" => "Entertainment"
                ]
            ]
        ],
        [
            "name" => "About"
        ],
        [
            "name" => "Contact"
        ]
    ];
    function showTieredMenu(array $menu) {
        echo "<ul>";
        foreach ($menu as $key => $item) {
            echo "<li>{$item['name']}</li>";
            if (array_key_exists('subMenu', $item)) {
                showTieredMenu($item['subMenu']);
            }
        }
        echo "</ul>";
    }
    showTieredMenu($menu);
?>
```

10. With the str function we can get information and transform of the str.
11. -
    a. The double quote mark will always interpreted new line escape string as a new line.
    b. While the single quotation mark will take things too literal.
    c. Here is the same case as point a.
    d. Here is the same case as point b.
    e. This case is similar to point a. but it is an example of indent escape string
    f. This case is similar to point b. but it is an example of indent escape string not being recognize.
    g. Here is an escape string meant to give clarity to the code that the part with \" is a literal double quotation mark string.
    h. This is a simalar case to point g. but it is protrayed in single quotation mark for string with single quotation mark.

12. The string message reversed.
13. It now reversed the word but did not change the sequence of the word in the sentence.
14. In my opinion, inline PHP is better because then you'll get the HTML emmet autocorrect instead of writing every element in HTML as strings.
15. The enitity with tag number is way more reliable than using the enitity alias.
16. The date function can be use to get the current date and format it to whatever you like. It also can be use to get the day in the week.
17. The date function can be us to get the current time in any timezone.

function.php

```php
<?php
// ! Experiment 1 - default function
// Membuat fungsi
// function perkenalan()
// {
//     echo "Assalamualaikum, ";
//     echo "Perkenalkan, nama saya Elok<br/>"; // Tulis sesuai nama kalian
//     echo "Senang berkenalan dengan Anda<br/>";
// }
//
// memamnggil fungsi yang sudah dibuat
// perkenalan();


// ! Experiment 2 - parametric function
// Membuat fungsi
// function perkenalan($name, $greeting)
// {
//     echo $greeting.", ";
//     echo "Perkenalkan, nama saya ".$name."<br>";
//     echo "Senang berkenalan dengan Anda<br>";
// }

// Memanggil fungsi yang sudah dibuat
// perkenalan("Hamdana", "Hallo");

```

```
27 // echo "<hr>";
28
29 // $saya = "Elok";
30 // $ucapanSalam = "Selamat Pagi";
31
32 // Memanggil lagi
33 // perkenalan($saya, $ucapanSalam);
34
35 // ! Experiment 3 - parametric function with default value
36 // Membuat fungsi
37 // function perkenalan($name, $greeting="Assalamualaikum")
38 // {
39 //     echo $greeting.", ";
40 //     echo "Perkenalkan, nama saya ".$name."<br>";
41 //     echo "Senang berkenalan dengan Anda<br>";
42 // }
43
44 // Memanggil fungsi yang sudah dibuat
45 // perkenalan("Hamdana", "Hallo");
46
47 // echo "<hr>";
48
49 // $saya = "Elok";
50 // $ucapanSalam = "Selamat Pagi";
51
52 // Memanggil lagi tanpa mengisi parameter salam
53 // perkenalan($saya);
54
55 // ! Experiment 4 - function with return value
56 // // Membuat fungsi
57 // function hitungUmur($tahun_lahir, $tahun_sekarang) {
58 //     $umur = $tahun_sekarang - $tahun_lahir;
59 //     return $umur;
60 // }
61
62 // echo "Umur saya adalah ". hitungUmur(2004, 2023) ." tahun";
63
64 // ! Experiment 5 - function in a function
65 // membuat fungsi
66 // function hitungUmur($tahun_lahir, $tahun_sekarang) {
67 //     $umur = $tahun_sekarang - $tahun_lahir;
68 //     return $umur;
```

```
// }

// function perkenalan($name, $greeting="Assalamualaikum")
// {
//     echo $greeting.", ";
//     echo "Perkenalkan, nama saya ".$name."<br>";
//     echo "Saya berusia ". hitungUmur(2004, 2023) ." tahun<br>";
//     echo "Senang berkenalan dengan Anda<br>";
// }
// perkenalan("Elok");
?>
```

recursive.php

```
<?php
// ! Experiment 6 - recursive function
// function tampilkanHaloDunia() {
//     echo "Halo dunia! <br>";
//     tampilkanHaloDunia();
// }

// tampilkanHaloDunia();

// ! Experiment 7 - recursive looping

// for-loop looping method
// for ($i=1; $i <= 25; $i++) {
//     echo "Perulangan ke-{$i} <br>";
// }

// recursive looping method
// function tampilkanAngka(int $jumlah, int $indeks = 1) {
//     echo "Perulangan ke-{$indeks} <br>";

//     if ($indeks < $jumlah) {
//         tampilkanAngka($jumlah, $indeks + 1);
//     }
// }

// tampilkanAngka(20);

// ! Experiment 8 - recursive tiered menu case study
// $menu = [
```

```
30 //     [
31 //         "name" => "Dashboard"
32 //     ],
33 //     [
34 //         "name" => "News",
35 //         "subMenu" => [
36 //             [
37 //                 "name" => "Touring",
38 //                 "subMenu" => [
39 //                     [
40 //                         "name" => "Beach"
41 //                     ],
42 //                     [
43 //                         "name" => "Mountain"
44 //                     ]
45 //                 ]
46 //             ],
47 //             [
48 //                 "name" => "Culinary"
49 //             ],
50 //             [
51 //                 "name" => "Entertainment"
52 //             ]
53 //         ]
54 //     ],
55 //     [
56 //         "name" => "About"
57 //     ],
58 //     [
59 //         "name" => "Contact"
60 //     ]
61 // ];
62
63 // function showTieredMenu(array $menu) {
64 //     echo "<ul>";
65 //     foreach ($menu as $key => $item) {
66 //         echo "<li>{$item['name']}</li>";
67 //     }
68 //     echo "</ul>";
69 // }
70
71 // showTieredMenu($menu);
```

```php

// ! Experiment 9 - recursive tiered menu case study code challenge
// $menu = [
//     [
//         "name" => "Dashboard"
//     ],
//     [
//         "name" => "News",
//         "subMenu" => [
//             [
//                 "name" => "Touring",
//                 "subMenu" => [
//                     [
//                         "name" => "Beach"
//                     ],
//                     [
//                         "name" => "Mountain"
//                     ]
//                 ]
//             ],
//             [
//                 "name" => "Culinary"
//             ],
//             [
//                 "name" => "Entertainment"
//             ]
//         ]
//     ],
//     [
//         "name" => "About"
//     ],
//     [
//         "name" => "Contact"
//     ]
// ];

// function showTieredMenu(array $menu) {
//     echo "<ul>";
//     foreach ($menu as $key => $item) {
//         echo "<li>{$item['name']}</li>";
//         if (array_key_exists('subMenu', $item)) {
//             showTieredMenu($item['subMenu']);
```

```
114 //         }
115 //     }
116 //     echo "</ul>";
117 // }
118
119 // showTieredMenu($menu);
120 ?>
```

string.php

```
1  <?php
2  // ! Experiment 10 - string function
3  // $loremIpsum = "Lorem ipsum dolor sit amet consectetur adipisicing
   ↪  elit. Ad neque aliquam mollitia obcaecati, maxime omnis
   ↪  consequuntur animi ducimus dignissimos, nisi perspiciatis magnam
   ↪  consequatur dolore corporis.";
4  // echo "<p>{$loremIpsum}</p>";
5  // echo "Panjang Karakter: " . strlen($loremIpsum) . "<br>";
6  // echo "Panjang Kata: " . str_word_count($loremIpsum) . "<br>";
7  // echo "<p>" . strtoupper($loremIpsum) . "</p>";
8  // echo "<p>" . strtolower($loremIpsum) . "</p>";
9
10 // ! Experiment 11 - escape character
11 // echo "Baris\nbaru <br>"; // soal 10.a
12 // echo 'Baris\nbaru <br>'; // soal 10.b
13 // echo "Halo\nDunia <br>"; // soal 10.c
14 // echo 'Halo\nDunia <br>'; // soal 10.d
15
16 // echo "<pre>Halo\tDunia!</pre>"; // soal 10.e
17 // echo '<pre>Halo\tDunia!</pre>'; // soal 10.f
18
19 // echo "Katakanlah \"Tidak pada narkoba!\" <br>"; // soal 10.g
20 // echo 'Katakanlah \'Tidak pada narkoba!\' <br>'; // soal 10.h
21
22 // ! Experiment 12 - string reversing
23 // $message = "Saya arek Malang";
24 // echo strrev($message) . "<br>";
25
26 // ! Experiment 13 - string reversing per word
27 // $message = "Saya arek Malang";
28 // $messagePerWord = explode(" ", $message);
29 // $messagePerWord = array_map(fn($message) => strrev($message),
   ↪  $messagePerWord);
```

```php
// $message = implode(" ", $messagePerWord);

// echo $message , "<br>";
?>
```

enitities.html

```html
<!-- ! Experiment 14 - HTML entities -->
<!DOCTYPE html>
<html>
    <head>
        <title>Entities HTML</title>
    </head>
    <body>
        <p>It&#39; time to read a HTML5 book</p>
        <p>    Keuntungan dari menggunakan nama entities: Sebuah nama entitas mudah diingat. Kerugian dari menggunakan nama entities: Browser mungkin tidak mendukung semua nama entitas, tetapi dukungan untuk nomor lebih baik.</p>
        <p>&#169; 2023 jti.com</p>
    </body>
</html>
```

date.php

```php
<!-- // ! Experiment 15 - date function -->
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <h3> Date </h3>
        <?php
            echo "Today is " . date("Y/m/d") . "<br>";
            echo "Today is " . date("Y.m.d") . "<br>";
            echo "Today is " . date("Y-m-d") . "<br>";
            echo "Today is " . date("l");
        ?>
    </body>
</html>
```

time.php

```php
<!-- // ! Experiment 16 - date function for time -->
<!DOCTYPE html>
<html>
    <head>
    </head>
    <body>
        <h3> Time </h3>
        <?php
            date_default_timezone_set("asia/jakarta");
            echo date("h:i:sa");
        ?>
    </body>
</html>
```