

Funkcje tekstowe

JavaScript i PHP – porównanie i przykłady

Dodawanie zmiennych do tekstu: konkatenacja i interpolacja

łączenie stringów i zmiennych

„wtrącanie” zmiennych w string

JavaScript

```
let txt = "ma"  
console.log("Ala " + txt + " kota")
```

```
let wiek = 25  
let napis = `Mam ${wiek} lat`
```

PHP

```
$txt = "ma";  
echo "Ala " . $txt . " kota";
```

```
$wiek = 25;  
echo "Mam $wiek lat";
```

Uwaga na apostrofy i cudzysłowy!

```
echo 'Mam $wiek lat';
```

Długość łańcucha tekstowego .length i strlen()

JavaScript

```
let txt = "lorem"  
console.log(txt.length) // 5
```

```
let longText = "The quick brown fox"  
console.log(longText.length)  
// 19 (16 + 3 spacje)
```

PHP

```
$txt = "ipsum";  
echo strlen($txt); // 5
```

```
$longText = "jumps over the lazy dog";  
echo strlen($longText);  
// 23 (19 + 4 spacje)
```

Pobieranie znaku na danej pozycji

JavaScript

```
let txt = "Gandalf the Grey"  
// poprzez odwołanie jak do tablicy  
console.log(txt[2]) // "n"  
// poprzez odpowiednią metodę  
console.log(txt.charAt(3)) // "d"
```

PHP

```
$txt = "Gandalf the White";  
// PHP nie ma takiej funkcji, ale jest  
na to obejście używające funkcji  
strpos()  
// dla prostych tekstów można  
odwołać się jak do tablicy  
$echo $txt[12]; // "W"  
$echo $txt[13]; // "h"
```

Zamiana wielkości liter

.toUpperCase(),
strtoupper(),

JavaScript

```
let txt = "JavaScript"  
// zamiana na wielkie litery  
console.log(txt.toUpperCase())  
// "JAVASCRIPT"  
  
// zamiana na małe litery  
console.log(txt.toLowerCase())  
// "javascript"
```

.toLowerCase(),
strtolower()

PHP

```
$txt = "PHP: HyperText Preprocessor";  
// zamiana na wielkie litery  
echo strtoupper($txt);  
// "PHP: HYPERTEXT PREPROCESSOR"  
  
// zamiana na małe litery  
echo strtolower($txt);  
// "php: hypertextpreprocessor"
```

Pobieranie pozycji tekstu .indexOf(), strpos()

JavaScript

```
let txt = "pomarańczowy"  
// zwraca indeks miejsca, od którego napis  
// się rozpoczyna  
console.log(txt.indexOf("ra")) // 4  
// lub -1, jeśli napisu nie znaleziono  
console.log(txt.indexOf("x")) // -1  
// UWAGA – zwracana jest pozycja  
pierwszego wystąpienia napisu  
// istnieje metoda lastIndexOf, która zwraca  
// pozycję ostatniego wystąpienia napisu
```

PHP

```
$txt = "niebieski";  
// zwraca indeks miejsca, od którego napis  
// się rozpoczyna  
echo strpos($txt, "b"); // 3  
// można też tak  
$szukana = "s";  
echo strpos($txt, $szukana); // 6  
// ofc w JS też można użyć zmiennej w stylu  
$szukana
```

Znajdowanie tekstu w tekście

.includes(), str_contains()

JavaScript

```
let txt = "herbata"  
console.log(txt.includes("bat")) // True  
// tu też można posłużyć się zmienną  
let look = "man"  
console.log(txt.includes(look)) // False
```

PHP

```
$txt = "ziołowa";  
echo str_contains($txt, "zio") // True  
// tu też można posłużyć się zmienną  
$look = "mek";  
echo str_contains($txt, $look); // False  
// szukanie pustego stringa "" zawsze  
zwraca True
```

Sprawdzanie początku/końca tekstu

<code>.startsWith()</code>	<code>str_starts_with()</code>
<code>.endsWith()</code>	<code>str_ends_with()</code>

JavaScript

```
let txt = "w3schools.com"  
console.log(txt.startsWith("w3")) // True  
console.log(txt.startsWith("www")) // False  
// tu też można posłużyć się zmienną  
let end = ".com"  
  
console.log(txt.endsWith(end)) // True
```

PHP

```
$txt = "php.net";  
echo str_starts_with($txt, "php") // True  
echo str_starts_with($txt, "js"); // False  
// tu też można posłużyć się zmienną  
$end = ".com";  
  
echo str_ends_with($txt, $end); // False
```


„Wyjęcie” podciągu znakowego .substring(), substr()

JavaScript

```
let txt = "Aphex Twin"  
  
// argument określający „początek” podciągu jest  
obowiązkowy  
  
// argument określający „koniec” podciągu jest  
opcjonalny  
  
console.log(txt.substring(7)) // "win"  
console.log(txt.substring(0, 3)) // "Aph"  
  
// można operować na ujemnych indeksach  
console.log(txt.substring(-4)) // "Twin"  
  
// w JS istnieje też metoda .slice(), która działa  
niemal identycznie
```

PHP

```
$txt = "Tangerine Dream";  
  
// argument określający „początek” podciągu jest  
obowiązkowy  
  
// argument określający „koniec” podciągu jest  
opcjonalny  
  
echo substr($txt, 12); // "eam"  
echo substr($txt, 1, 6); // "anger"  
  
// można operować na ujemnych indeksach  
echo substr($txt, -5); // "Dream"
```

Podział napisu na tablicę .split(), explode()

JavaScript

```
let txt = "Fight Club"  
console.log(txt.split(" "))  
// ["Fight", "Club"]  
// można operować na „własnym” znaku  
rozgraniczającym  
let innyTxt = "Spec Ops: The Line"  
console.log(innyTxt.split(":"))  
// ["Spec Ops", " The Line"]
```

PHP

```
$txt = "Behind Enemy Lines";  
// podanie znaku rozgraniczającego jest  
obowiązkowe (nie może to być pusty string "")  
$podzielone = explode(" ", $txt);  
// ["Behind", "Enemy", "Lines"]  
$innyTxt = "Fullmetal Alchemist: Brotherhood";  
$innePodzielone = explode(": ", $innyTxt);  
// ["Fullmetal Alchemist", "Brotherhood"]
```

Złączenie tablicy w napis

.join(), implode()

JavaScript

```
let tab = ["Cas", "io", "pe", "a"]  
console.log(tab.join()) // "Casiopea"  
// można operować na „własnym” znaku  
// rozgraniczającym  
let innyTab = ["ba", "a"]  
console.log(innyTab.join("na"))  
// "banana"
```

PHP

```
$tab = ["Na", "ga", "nu", "ma"];  
echo implode($tab);  
// "Naganuma"  
// można operować na „własnym” znaku  
// rozgraniczającym  
$innyTab = ["Men", "Without", "Hats"];  
echo implode(" ", $innyTab);  
// "Men Without Hats"
```

Zamiana fragmentu tekstu na inny

.replace(), str_replace()

JavaScript

```
let txt = "samoloty"
let zamieniony = txt.replace("lot", "chod")
console.log(zamieniony) // "samochody"
// domyślnie ta metoda zamienia tylko pierwsze
// wystąpienie danego napisu
// aby zamienić każde, należy użyć wyrażenia
// regularnego (RegExp) z dopiskiem /g - global
let innyTxt = "Rozpieszczony pies"
let innyZamieniony = innyTxt.replace(/pies/g, "kot")
console.log(innyZamieniony) // "Rozkotchczony kot"
// ...albo użyć .replaceAll()
```

PHP

```
$txt = "niebezpieczne";
echo str_replace("e", "o", $txt)
// "niobozpioczno"
// można doprecyzować, ile zamian
// wykonać
$innyTxt = "ananas";
echo str_replace("a", "y", $innyTxt, 2);
// "ynynas"
```

Źródła:

- <https://kursjs.pl/kurs/super-podstawy/string>
- https://www.w3schools.com/php/php_ref_string.asp

Zadania JavaScript

1. Generator hasła - napisz kod, który przyjmuje ciąg znaków i generuje hasło na jego podstawie:
 - usuń wszystkie spacje z ciągu
 - zamień wszystkie litery na duże
 - wyciągnij pierwsze 8 znaków jako wynikowe hasło
 - jeśli tekst jest krótszy niż 8 znaków, dodaj do niego losowe cyfry, aby miał długość 8
 - wypisz hasło na stronie, obok pola formularza, w którym wpisywano ciąg znaków
 - przykładowo, dla ciągu „Małgorzata”, wynikiem będzie „MAŁGORZA”, a dla ciągu „Jan”, wynikiem może być „JAN02241”
2. Weryfikacja zdania – napisz kod, który przyjmuje dowolne zdanie i:
 - sprawdza, czy zdanie zaczyna się od dużej litery (czy znak na pierwszym miejscu jest taki sam jak znak na pierwszym miejscu, jeśli zrobimy z niego wielką literę?)
 - sprawdza, czy zdanie kończy się wykrzyknikiem
 - jeśli oba warunki są spełnione, napisz obok pola formularza, w którym wpisywano zdanie, napis „Poprawne :)”, w przeciwnym przypadku „Niepoprawne :(”
 - przykładowo, dla zdania „Obcy: Ósmy Pasażer Nostromo”, wynikiem będzie „Niepoprawny :(”, a dla zdania „Whiplash!”, wynikiem będzie „Poprawny :)”

Zadania JavaScript – cd.

3. Wyszukiwanie i podmiana w zdaniu - napisz kod, który przyjmuje dwa argumenty: zdanie i słowo do wyszukania. Kod powinien:
 - sprawdzić, czy słowo znajduje się w zdaniu
 - jeśli tak, zastąpić je napisem „[REDACTED]”
 - podzielić zdanie na tablicę słów
 - połączyć tablicę z powrotem w zdanie oddzielone myślnikami i wyświetlić je obok pola formularza, z którego pobierane jest zdanie
 - przykładowo, dla zdania „JavaScript jest łatwy” i słowa „łatwy”, wynikiem będzie „JavaScript-jest-[REDACTED]”
4. Analiza listy e-maili – napisz kod, który dla ciągu adresów e-mail oddzielonych przecinkami, np.: „user1@gmail.com, user2@yahoo.com, user3@hotmail.com”:
 - podzieli e-maile na tablicę
 - dla każdego elementu tablicy, sprawdź, czy domena to „gmail.com”
 - jeśli tak, zmień wszystkie litery tego e-mailu na wielkie
 - połącz zmodyfikowaną tablicę z powrotem w jeden ciąg, oddzielając elementy średnikami
 - dla powyższej przykładowej listy e-maili, wynikiem będzie: „USER1@GMAIL.COM; user2@yahoo.com; user3@hotmail.com”

Zadania PHP

wyniki obu tych zadań należy wyświetlić na innych podstronach

5. Automatyczne formatowanie zdania - napisz kod, który przyjmuje ciąg znaków i:
 - sprawdza, czy zdanie zaczyna się wielką literą i kończy kropką
 - jeżeli któryś z warunków jest niespełniony, należy automatycznie poprawić zdanie
 - liczy i wyświetla ilość słów w zdaniu (do policzenia słów należy użyć funkcji count() na tablicy powstałej po rozbiciu zdania)
 - przykładowo, dla zdania „you shall not pass”, wynikiem będzie „You shall not pass.” oraz liczba 4
6. Usuwanie słów zakazanych – napisz kod, który przyjmuje dowolne zdanie i:
 - sprawdza, czy w zdaniu znajduje się jakieś ze słów „zakazanych” z ustalonej wcześniej listy, np.: [„jabłko”, „banan”] (trzeba stworzyć pętlę, która będzie wykonywać się tyle razy, ile jest zakazanych słów)
 - jeżeli tak, zamienia je na „***” i wyświetla zamienione zdanie
 - przykładowo, dla powyższych „zakazanych” słów i zdania „Shake bananowy z jabłkowym sokiem”, wynikiem będzie: „Shake ***owy z ***wym sokiem”