

JS 04 – Szyfrowanie (22)

Napis (łańcuch tekstowy), to ciąg znaków ujęty w cudzysłowy lub apostrofy: "Wacek" lub 'Wacek Libront'.

Napisem jest też tzw. łańcuch pusty: '' lub ''''.

Znaki w napisach są ponumerowane (indeksowane). Pierwszy znak w napisie ma numer zero.

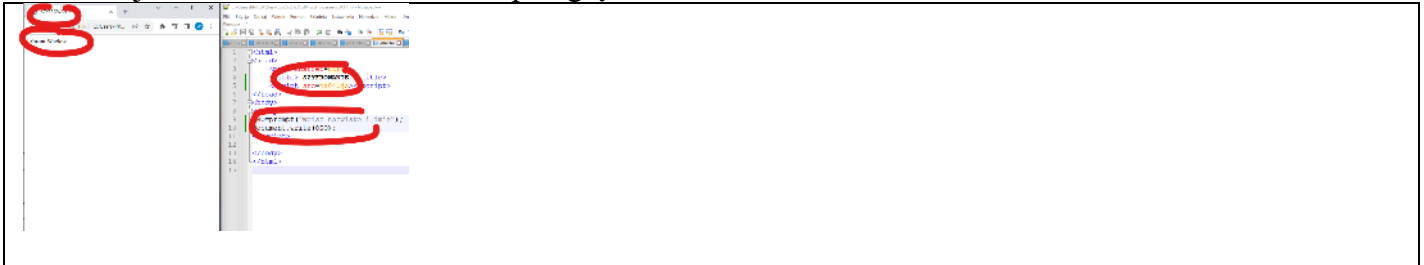
Dostęp do pojedynczych znaków realizujemy za pomocą nawiasów [].

Pliki (2)

- W swoim folderze utwórz 2 nowe dokumenty: **js04.html** i **js04.js**
- Otwórz oba dokumenty w notatniku, a dokument HTML w przeglądarce
- Dokumentu **HTML** wklej tekst z ramki

```
<html>
<head>
  <meta charset=utf8>
  <title> SZYFROWANIE </title>
  <script src=js04.js></script>
</head>
<body>
<script>
  OSO=prompt("Wpisz swoje nazwisko i imię");
  document.write(OSO);
</script>
</body>
</html>
```

- Zmień tytuł strony **SZYFROWANIE** na swoje **inicjały**
- Zapisz dokumenty i odśwież przeglądarkę
- Wpisz swoje nazwisko i imię
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML



Funkcje (2)

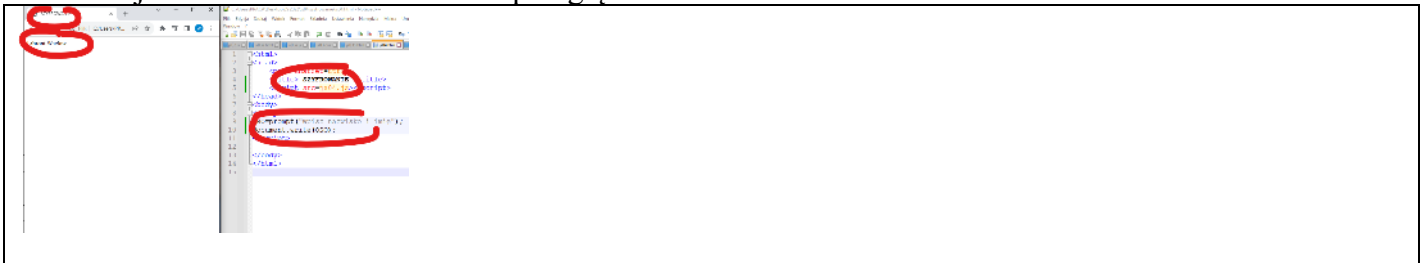
- Do dokumentu **JS** wpisz tekst z ramki

```
function BR() {document.write("<br>");}
function WR(napis) {document.write(napis);}
```

funkcja BR() – wypisuje na stronie znacznik przejścia do nowego wiersza zamiast document.write(„
”);

funkcja WR(napis) – wpisuje na stronie napis zamiast document.write(napis);

- Za pomocą tych funkcji
 - napisz na stronie **Twoje nazwisko**
 - napisz pod spodem **Twoje imię**
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML



Funkcja Przystawny (2)

Ten typ szyfrowania przestawia znaki w innej kolejności. Najprostszym przykładem takiego szyfrowania jest pisanie wspak. Natomiast funkcja Przystawny() zamienia miejscami dwa kolejne znaki napisu podanego jako parametr.

- Do dokumentu JS wpisz tekst z ramki

```
function Przystawny(n) {  
    var d=n.length;  
    var w="";  
    for (var i=0;i<d-1;i=i+2){  
        var z1=n[i];  
        var z2=n[i+1];  
        w=w+z2+z1;  
    }  
    if (d%2==1){  
        w=w+n[d-1];  
    }  
    return w;  
}
```

Pętla iteruje w następujący sposób: 0, 2, 4, 6, 8...

Z napisu N pobieramy dwa kolejne znaki Z1[i] i następny Z2[i+1]

Do napisu W doklejamy te znaki w zmienionej kolejności

Jeżeli napis N ma nieparzystą długość, to

na końcu napisu W doklejamy ostatni znak napisu N

- Wykonaj inspekcję dokumentu JS - sprawdź, czy nie ma błędów
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu JS z funkcją

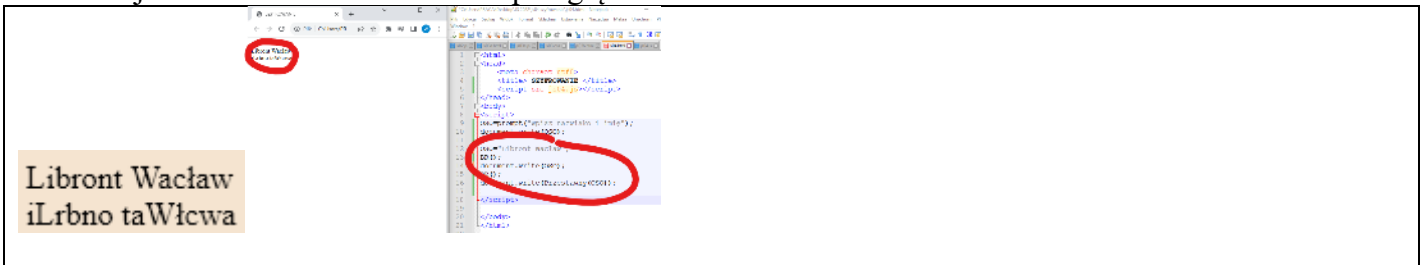


Szyfr przestawny (2)

- Do dokumentu HTML, przed znacznik </script> wklej tekst z ramki

```
BR();  
WR(OSO);  
BR();  
WR(Przystawny(OSO));
```

- Zapisz dokumenty i odśwież stronę
- Wpisz swoje nazwisko i imię
na ekranie powinno pojawić się „przestawione” Twoje nazwisko i imię
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML



Funkcja Przesun (2)

Szyfr przestawieniowy stosowany już w starożytności, w którym każdy znak w napisie jest zastępowany przez inny, oddalony o stałą liczbę pozycji (tzw. klucz) w zestawie znaków (tzw. alfabecie). Zmienna ALF jest alfabetem i wpisana jest w postaci tekstu i będziesz ją mógł wkleić w swój program.

ALF="AĄBCĆDDEĘFGHIJKLLMNNŃOÓPQRSSTUVWXYZŻaąbcćdeęfghijklmnnóopqrrsstuvwxzyżż "

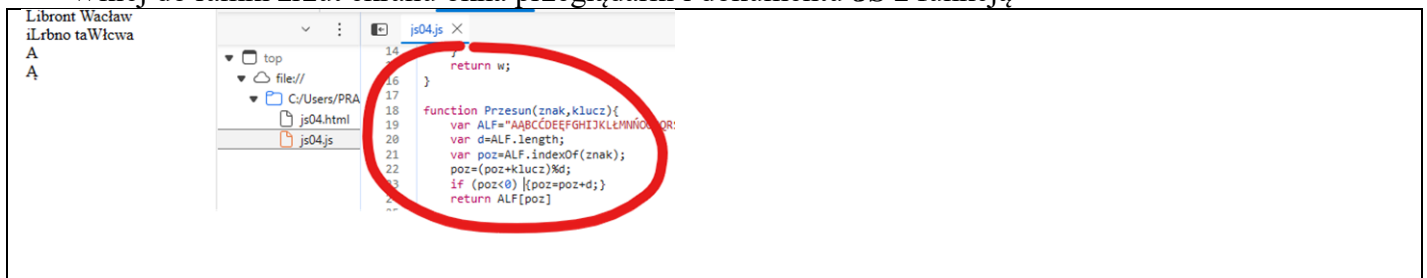
- Do dokumentu JS wpisz tekst z ramki

- Do zmiennej ALF skopiuj cały napis

```
function Przesun(znak,klucz){
    var ALF="AABCDCDEEFGHIJKLŁMNŃOÓPQRSSTUVWXYZŻżaaabccdeēfghijklłmnñoópqrsstuvwxyzżż ";
    var d=ALF.length;
    var poz=ALF.indexOf(znak);
    poz=(poz+klucz)%d;
    if (poz<0) {poz=poz+d;}
    return ALF[poz]
}
```

Funkcja Przesun() przesuwa ZNAK w alfabecie ALF o KLUCZ miejsc
 Za pomocą polecenia indexOf() znajduje pozycję znaku w napisie ALF
 do pozycji POZ dodaje KLUCZ i wykonuje operację modulo
 gdy nowa pozycja jest dłuższa niż ALF, to zliczanie od początku alfabetu
 jeżeli pozycja jest ujemna, to powiększ o długość alfabetu – uwzględniamy ujemny klucz
 funkcja zwraca znak z nowej pozycji alfabetu ALF

- Wykonaj inspekcję dokumentu JS - sprawdź, czy nie ma błędów
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu JS z funkcją

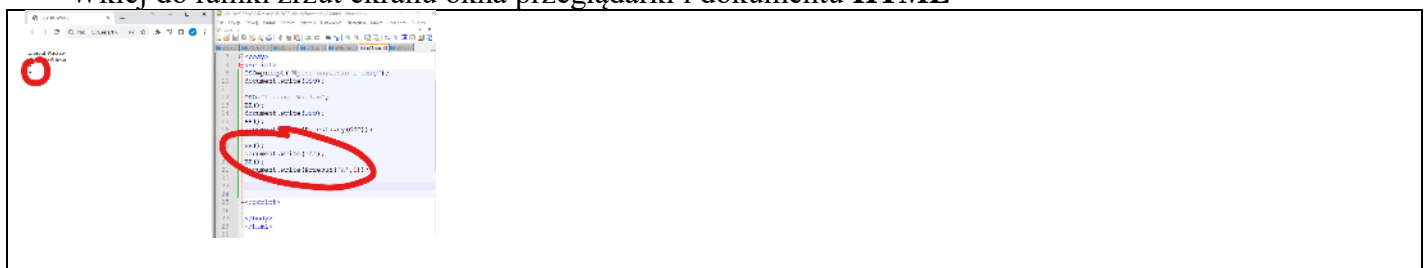


Przesuwanie znaków (2)

- Do dokumentu **HTML**, przed znacznik </script> wklej tekst z ramki

```
BR();
WR('A');
BR();
WR(Przesun('A',1));
```

- Zapisz dokumenty i odśwież stronę
- Wpisz swoje nazwisko i imię
instrukcje przesuwać znak 'A' o jedną pozycję
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu **HTML**



Funkcja Cezar (2)

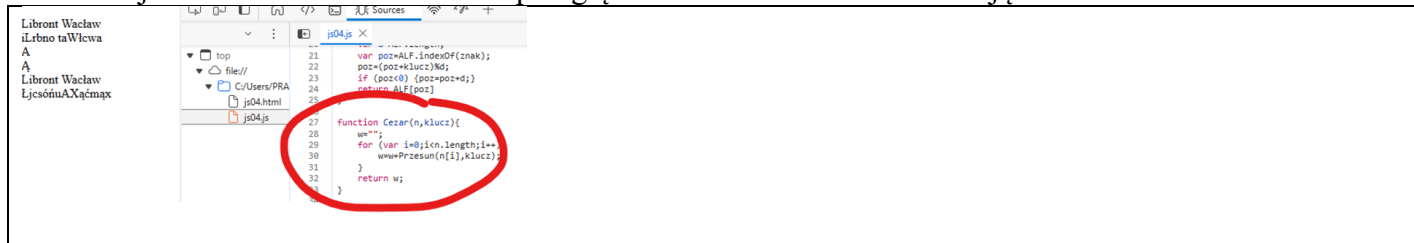
Nazwa szyfru pochodzi od Juliusza Cezara, rzymskiego wodza i polityka. Szyfrował on prywatną korespondencję do swoich przyjaciół, zapisaną po łacinie, używając szyfru przesuwanego z kluczem 3. W typowych zastosowaniach szyfr korzysta z tzw. tablicy kodów ASCII, my posłużymy się alfabetem w zmiennej ALF, która jest wpisana w postaci tekstu i będziesz ją mógł wkleić w swój program.

- Do dokumentu **JS** wpisz tekst z ramki

```
function Cezar(n,klucz) {
    w="";
    for (var i=0;i<n.length;i++){
        w=w+Przesun(n[i],klucz);
    }
    return w;
}
```

Funkcja CEZAR przesuwa każdą literę napisu *N* o liczbę miejsc opisaną w parametrze *KLUCZ*
 pętla FOR wybiera kolejne znaki z napisu
 przesunięty znak zostaje „wyliczony” za pomocą funkcji PRZESUN
 przesunięte znaki sklejane są w łańcuch tekstowy w zmiennej *W*

- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu JS z funkcją

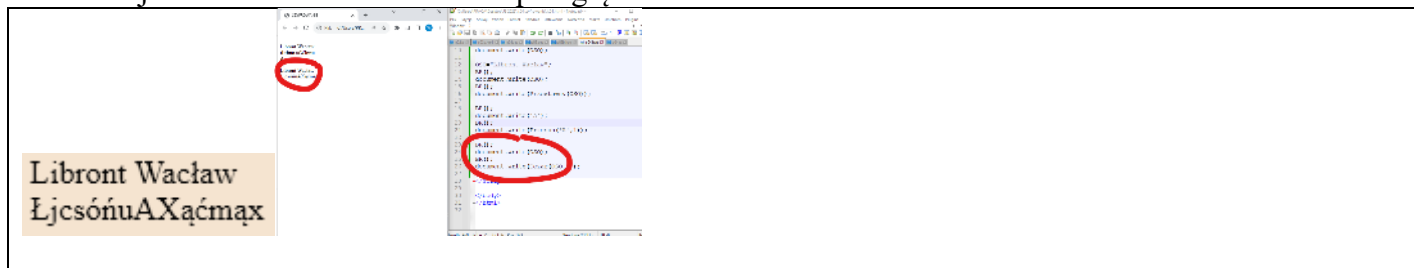


Szyfr Cezara (2)

- Do dokumentu **HTML**, przed znacznik `</script>` wklej tekst z ramki

```
BR();
WR(OSO);
BR();
WR(Cezar(OSO,1));
```

- Zapisz dokumenty i odśwież stronę
- Wpisz swoje nazwisko i imię
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu **HTML**



Liczby losowe (2)

Do testowania sortowania potrzebne są nieuporządkowane dane liczbowe. Zamiast wpisywać je ręcznie posłużymy się generatorem liczb losowych. Funkcja `Losuj()` zwraca losową liczbę z określonego przedziału.
 Listy są strukturami danych, które mogą przechowywać wiele wartości tego samego typu. To taka tablica, z komórkami na przykład na liczby.

- Do dokumentu **JS** wpisz tekst z ramki

```
function Losuj(pocz, koni) {
    return Math.floor(Math.random() * (koni-pocz+1)+pocz);
}
```

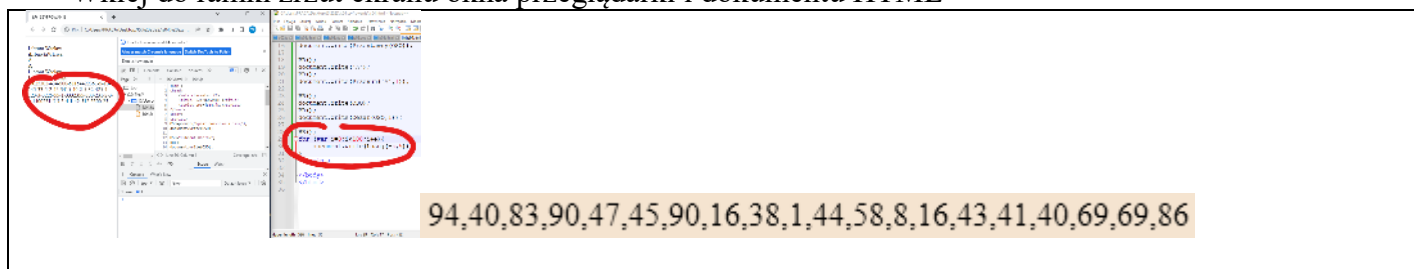
Funkcja `MATH.RANDOM` daje rzeczywistą liczbę losową z przedziału `0..1`
 liczba jest mnożona przez „rozpiętość” liczb `(KONI-POCZ+1)` i przesunięta o `POCZ`
 a następnie zamieniana na liczbę całkowitą

- Do dokumentu **HTML**, przed znacznik `</script>` wpisz tekst z ramki

```
var L=[];
for (var i=0;i<20;i++){
    L.push(Losuj(1,100));
}
BR();
WR(L);
```

Losujemy 20 liczb z przedziału $\langle 1..100 \rangle$ do tablicy L
 Deklarujemy tablicę L początkowo pustą $[\]$
 Za pomocą funkcji $PUSH$ dodajemy nowy element na koniec tablicy

- Zapisz dokumenty i odśwież stronę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML



Funkcja SortBab (2)

Sortowanie i wyszukiwanie informacji zajmuje współczesnym komputerom znaczą część czasu pracy i szybkość algorytmów wpływa znacząco na jakość pracy z programami. Wyobraźmy sobie, gdyby wyszukiwarka Google podawała wynik po 10 minutach w nieuporządkowany sposób!

Sortowanie bąbelkowe jest jednym z najpopularniejszych sposobów sortowania - porównujemy dwa kolejne elementy listy, jeżeli nie są uporządkowane, to zamieniamy je miejscami.

- Do dokumentu **JS** wpisz tekst z ramki

```
function SortBab(lista){
  var d=lista.length;
  for (var i=0;i<d;i++){
    for (var j=0;j<d;j++){
      if (lista[i]<lista[j]){
        var l=lista[i];
        lista[i]=lista[j];
        lista[j]=l
      }
    }
  }
  return lista;
}
```

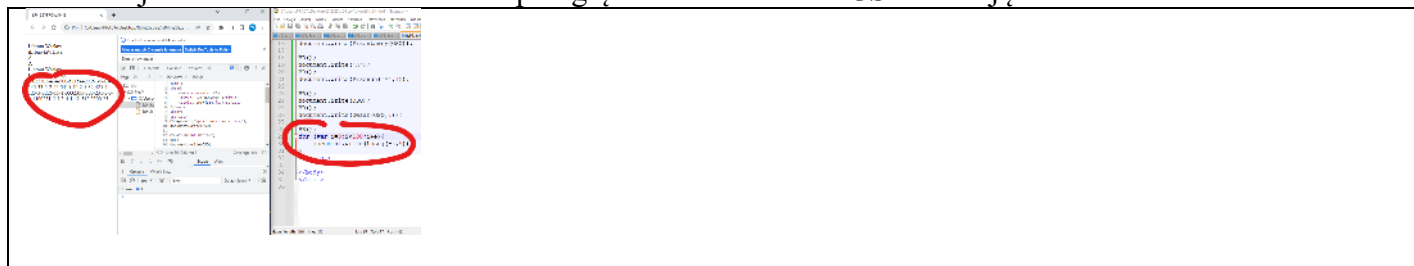
Funkcja $SortBab$ sortuje podaną jako parametr listę – tablicę liczb

Pętla wewnętrzna (ze zmienną J) porównuje kolejne elementy listy i największy przesuwa na koniec

Pętla zewnętrzna (ze zmienną I) uruchamia pętlę wewnętrzną tyle razy ile elementów ma lista – za każdym razem jedna liczba przesuwana

Zamieniamy miejscami dwa elementy na liście za pomocą dodatkowej zmiennej L

- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu **JS** z funkcją



Bąbelki (2)

- Do dokumentu **HTML**, przed znacznik $\langle /script \rangle$ wklej tekst z ramki

```
BR();
WR(L);
BR();
WR(SortBab(L));
```

wylosowane liczby w zmiennej L są sortowane bąbelkowo

- Zapisz dokumenty i odśwież stronę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML

