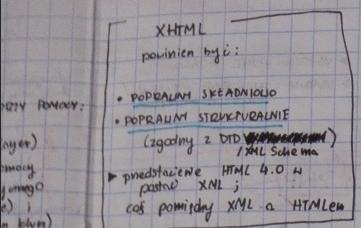


PHP DALEJ



DYNAMIKA STRONY

- HTML + CSS - static
- STRONA Klienta (Przeglądarka)
- JAVASCRIPT
mod trafi w przeglądarkę
- PO STRONIE SERWERA
- PHP, Python, ...
w generowane HTML przez aplikacji dzierżące po stronie serwera
- ASYNCHRONICZNE
- AJAX
dynamiczne, aby pobrać dane z serwera i generować

- DOM (Document Object Model) -

- reprezentacja dokumentu w celu modyfikacji
- mą postać drzewa elementów
- API DOM - interfejsy do operowania na danej stronie dokumentu
- pozwala na wyszukiwanie, modyfikowanie i dodawanie el. dokumentu

① WYSZUKIWANIE

NAZWA → document.getElementsByTagName('div')
znacznika NAME → getElementsByName('name')
CLASS → getElementsByClassName('classname')

PÓZDYNKI, ID → getElementById('footer')

PÓZDYNKI, ŚWIĘCKA → querySelector('div#body div.p')
ZBIÓR EL., ŚWIĘCKA → querySelectorAll('div #body div.p h1')

- #### ② USUŁYANIE
- wysułuje el.: var delete = document.getElementById('...');
 - delete: parentNode.removeChild(delete);
↑ usuwa, usuwa dziecko rodzeństwo

③ DODAWANIE NOVYCH

```
var div = document.createElement('div');
var text = document.createTextNode('tekst');
div.appendChild(text);
document.body.appendChild(div);
```

JAVASCRIPT

- mod struktury i treści po załadowaniu strony [API DOM]
- reakcja na działania użytkownika (np. click button)
- Walidacja formularza bez wysyłki na serwer
- wyszczególnienie reklam
- komunikacja synchroniczna [AJAX]

GDZIE?

zazn. plik
eg. <script src="..."/>
w pliku HTML
wykonane przed KIEDY?
zatłodowaniem dokumentu HTML

w otrymującym obiektu
zdarzeniu w zmianie
eg. <button onclick="...">

wymiana przy zdarzeniu
np. hover, click

window.onload - po załadowaniu całego docu HTML
i wszystkich zasobów (obrazów etc.)

SKŁADNIA

*! DEKLARACJA ZMIENIEN:

(nie ma typów)

var nazwa = "wartość";

* FUNKCJE

```
function nazwa(zmienna){
    return 'K' + zmienna;
}
```

← TAKIEJEME STRINGÓŁKI

* TAB:

var znaki = ['a', 'b', '15']; ← różne typy danych
var tab = [znaki, ['text', 5, 'c']] ← wielowymiarowe

*! BRAK DEKLARACJI ROZMIARU - DYNAMICZNIE, ROZSZERZALNE
znaki[znaki.length] = 3; ← doda nowy el. na końcu

* OBIEKTY

```
var osoba = {
    imie: 'Alina',
    'nr': 333333
};
```

osoba.imie = osoba['imie']

można dodać funkcje w obiekcie

JSON (JavaScript Object Notation)
object JS → text
syntaksa parseable
format wykorzystywany w innych

```
var str = JSON.stringify(osoba),
    var object = JSON.parse(str);
```

WEB STORAGE API

- wszystkie przechowywane dane
- w postaci tekstowej!

W PRZEGŁĄDARCE

- | | |
|--------------------------------------|---------------------------------------|
| LOCAL STORAGE | SESSION STORAGE |
| dostępne też po zmianie przeglądarki | znikają po zamknięciu otwartej sesji! |
| → no expiration date | → cookies |

COOKIES

→ MAJE DOBÓR DANYCH

→ PRZEŚLANY NA SERWER

→ W PRZEGŁĄDARCE

PROTOKOLUM

WEB STORAGE

→ SESSIONStorage, localStorage

→ WYSZCZEGÓLNIENIA

(D. 1000 MB)

→ DANE LOKALNE

PHP DALEJ

* KONIEC PRZETWARZANIA SKRYPTU:

exit(); die();

* TABLICE

array

zmienna rozmiar jak w JS: \$imiona[] = 'Bart';

e.g. \$imiona = array('Piotr', 'Tomasz', 'Marek')

\$imiona = ['Piotr', 'Tomasz', 'Marek', 'Sławek']
indeksowanie od 0

```
$studenci = [
    'Mark' => 20,
    'Kasia' => 21,
    'Asia' => 19
];
```

//litasne klucze

może mieć w sobie inne typy, wyciągać dającą jąc obiekty

```
$zdj = [
    'title' => 'A', 'autor' => 'Kamil',
    'title' => 'B', 'autor' => 'Nope',
];
```

* FUNKCJE ANONYMOUS

- wykorzystać do zmiennej instancji
- w ten sposób przekazywać jako arg do innej funkcji :D [callback]

ZADANIA W PHP

- GET • parametry w URL
 • pobieranie img, script etc., inf. 2
 • bezparametryczne
- POST • form post • parametry w deklaracjach
 • zmieniące stan strony zadania

FORMULARZ

* GET

strona.php?param=koły&imie=kula

\$GET['imie'] → kula
 dopasowane; pobierz inf. 2 adresu URL
 dostosuj do struktury parametru zapytanie

* POST

informuje o formularzu

\$POST['autor']

radio

text → name=""

password

<input type="text" name="autor">

checkbox → name="kategorie[]"

checkbox 2 zaznaczonymi

LACZENIE PLIKÓW

- include 'path.php';
 - jeśli nie istnieje → WARNING, ale kod dalej działa
- działa i wykonuje
- PLIK OPCJONALNY

* require 'plik.php';

- jeśli nie istnieje → Fatal error, kod stop

* PLIKI WYMAGANE

* include - once, require - once

- Tylko raz, kiedy nie chcemy przypadkowego podpisania zmieniających się.

FUNKCJE

function nazwa(\$zmienna)

{
 echo "\$zmienna". '
';
 }

& \$zmienna - przekazanie przez referencję

function dwa(\$imie='Ola', \$k='YAY') {

 //...
 }
 ↑
 ARGUMENTY DOMYSLNE,
 jeśli wywołane: dwa() to używa domyślnych

* ZMIENNE ZMIENNE

! \$k = "OK";
 \$hmm = "k";

echo \$\$hmm; //zwroć OK

IT CAN GO DEEPER

function pozdr(...){ ... }

\$wybrana = 'pozdr';
 echo \$wybrana(5); ← wywoła funkcję pozdr

VISUAL BASIC W FRAMEWORKACH

JAKIE ZADANIE NA SERWER?

* \$SERVER['REQUEST-METHOD']

↑ mówi my POST my GET

hidden - nie wyświetlony w przeglądarce, ale dostosowany do POST

disabled - niedostosowany do PHP!

checkbox 2 zaznaczonymi

FORMULARZE CD.

• UPLOAD PLIKÓW

enctype="multipart/form-data"

<input type="file" name="zdj"/>

\$FILES['zdj']

↑
inf. o wszystkich przesłanych plikach

CIASTECZKA

► USTALANE PRIĘD WYSŁANIEM CAŁEGO ODP
(BO WYSŁANE W NAGŁÓWKACH!)

• [setcookie ("nazwa_ciszka", \$wartosc_zapisana), time() + 3600, "/")
↑
ustawia cookie ozn. trwała / umieszcza zarket (tu: cała domena)

• [isset (\$COOKIE['nazwa_ciszka'])] ← sprawdza my jest true cookie

• USUŁANIE? → nadpisanie 2 razem umieszczonego mingT :D
→ zmiana wartości na false :D

MONGODB

kolejka - zbiór el. tego samego typu! ale mogą mieć różne struktury :D
jedna MongoDB może hostować wiele baz danych!

► 2API'S DO KOLEKCIJ

czyżby kolejka if "\$db" → produkty → insert(\$produkt);
nie ma)

► KAŻDY DOKUMENT W MongoDB ma pole -id

► wyszukiwanie: find() ← ZWRACA LISTY

find({query}) ← ZWRACA PASZTYCĘ
przykład doboru, trzymamy z np. tylko jednym polem - reżeta dowolna

w query ustalamy tylko -id

findOne({query}) ← ZWRACA JEDEN

skip(10) - pominięcie pierwszych 10
limit(3) - poza tym 3

bieżące 3

\$SESSION ← sprawdzanie domysh o sesji, czy jest i jaka wartość

NOPE: MDS SHA-1

DOBRE f. bcrypt password.hash()

password.verify()

ad PHP 5.5

WYLOGOWANIE: • usiąść dane sesji

POST - REDIRECT - GET

► KAŻDE POST, KTÓRE ENCRYPTOWAŁO I ZMIENIŁO, POWINNO WYSTAĆ REDIRECT (3XX)

PONIŻEJ Location

► WYSZYSTKIE NAGŁÓWKI USTALANE PRIĘD WYSŁANIEM ANSYGNAŁA HTML

► REQUEST - m.in. info o POST, GET;
COOKIE

• [setcookie ("nazwa_ciszka", \$wartosc_zapisana), time() + 3600, "/")
↑
ustawia cookie ozn. trwała / umieszcza zarket (tu: cała domena)

• [isset (\$COOKIE['nazwa_ciszka'])] ← sprawdza my jest true cookie

• USUŁANIE? → nadpisanie 2 razem umieszczonego mingT :D
→ zmiana wartości na false :D

MECHANIZMY SESJI

COOKIES

• zapisane po stronie przeglądarki
NIEBEDP. - TAKIM DOSTĘP ALE

SESSJA

• zapisywanie po stronie serwera
identyfikator - DEBUG, LOSOWY

4 COOKIES

• session_start() ← ZACZYNIA SESJĘ
• session_id() ← kiedyś odnośnik i formie stałe pole id!
• session_id() ← kiedyś odnośnik i formie stałe pole id!
• session_start() ← ZACZYNIA SESJĘ

► SORTOWANIE: sort(['price' => 1])
po cenie ROŚNĄCO

sort(['price' => -1])
MAŁEJSZO

► USUŁANIE: remove(\$query)

'\$gt' ← większe niż '\$or' => [\$_id], [\$_id]

'\$lt' ← mniejsze niż '\$or' => [\$_id], [\$_id]

session_destroy() ← usuwa dane

! NIE zamienia sesji i
NIE usuwa cookie

► trzeba nowe ustawić COOKIE

KONTROLA DOSTĘPU

Rejestracja: • FORM logowania → haszowane hasło → zapis do db → redirect!

• phewlogowanie haszt [znać hash!] Logowanie: • FORM → pobrany z POST → weryfikacja usera → DONE :)

zmienna ID sesji → redirect();

NOPE: MDS SHA-1

DOBRE f. bcrypt password.hash()

password.verify()

ad PHP 5.5

WYLOGOWANIE: • usiąść dane sesji

nowas JavaSc

• BIBLIOTEKA JS

- UZUATWIA WŁÓCIEJĄCE AJAJA
- UZUATWIA TWORZENIE ANIMACJI / EFEKTÓW

MVC

(Model View Controller)

- front-controller - pojedynczy punkt wejściowy do app. umieszcza kontroler na dostarczanej adresu
 - z \$ - GET

- routing - odizorowuje parametry z adresu na kontroler do wywołania (jakie podstrony chce klient?)

- dispatcher - buduje odp na podstawie wywołania i tego co dalej (redirekt? zapisanie?)

- kontrolery - kod obsługujący stron
- widoki - oddzielony HTML od logiki biznesowej

- model - tylko to co przedstawia pomieszczy widokiem (View) o kontrolerem (rezyta danych ukrysta)

- WYDzielona LOGIKA BIZNESOWA (obsługa db)
OD KONTROLERÓW

