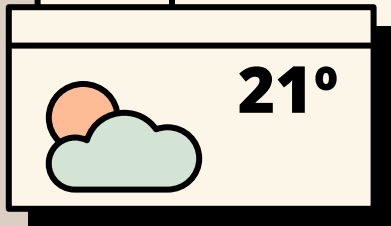
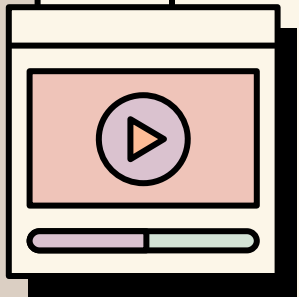


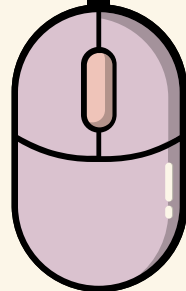
# Съхранение на локални данни в браузъра



Денислав Манахов

Мартин Менчев

Кирил Димов





**01** Web Storage API

**02** Cache

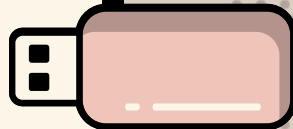
**03** WebSQL

**04** IndexedDB

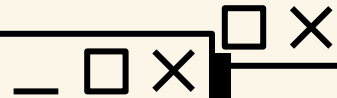
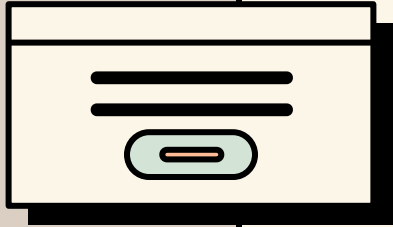
01

# WebStorage API

Как да съхраняваме данни в брауъра?



# Cookies



# Какво са бисквитките

Name	Value	Domain	Path	Expires ...	Size	HttpOnly	Secure
_gh_sess	HcBW0lIXvVOBkA023zHFIEG1nAzGk3j0rf...	github.c...	/	Session	386	✓	✓
_octo	GH1.1.1563279438.1715322853	.github.c...	/	2025-0...	32		✓
logged_in	no	.github.c...	/	2025-0...	11	✓	✓
preferred_color_mode	dark	.github.c...	/	Session	24		✓
tz	Europe%2FKiev	.github.c...	/	Session	15		✓

- Двойка от ключ и стойности
- Единственият начин за съхранение на информация преди HTML5
- Изпращат се със всяка HTTP заявка
- Могат да съхраняват информация с размери до 4KB
- Задава се „срок на годност“ за всяка бисквитка



# Видове бисквитки



## Постоянни

Имат срок на годност.  
Като те остават в  
браузъра докато не  
изтече срока им или не  
се изтрият ръчно от  
потребителя



## Сесийни

Те нямат срок на  
годност. Създават се и  
се съхраняват в  
браузъра до неговото  
затваряне или  
затваряне на раздела.

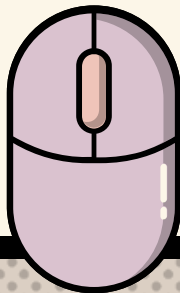


# Пример

```
document.cookie = "language=bg";  
document.cookie = "currency=BGN";  
  
console.log(document.cookie); // "language=bg; currency=BGN"  
  
document.cookie = "currency=; Max-Age=0" // Delete currency  
  
console.log(document.cookie); // "language=bg;"
```

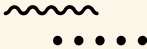
.....

>>>>





# Local Storage





# Какво е Local Storage

http://127.0.0.1:5500	
Origin http://127.0.0.1:5500	
Key	Value
userData	smartUser
shoppingCart	["Flip Flops", "Flip Flops", "Sunglasses", "Beach Ball", "Life Jacket", "Beach Ball"]

- Двойка от ключ и стойност
- Работи само върху браузъри с HTML5
- Може да съхранява информация с размери около 10MB
- Информацията се запазва за неопределено време



# Често използвани методи



## **setItem(key, value)**

Добавя или заменя дадения ключ със зададената стойност.

## **getItem(key)**

Връща стойността на даден ключ. Ако този ключ не съществува, връща null.

## **removeItem(key)**

Премахва елемент съдържащ се в LocalStorage с дадения ключ.

## **clear()**

Премахва цялата информация от LocalStorage

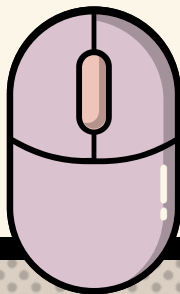


# Пример

```
localStorage.setItem("theme", "light");  
const theme = localStorage.getItem("theme");  
console.log(theme) // "light"
```

```
localStorage.removeItem("theme");  
localStorage.clear();
```

.....



# Session Storage



# Какво е Session Storage

Origin http://127.0.0.1:5501	
Key	Value
username	vankata69
age	42
lastName	Kolev
firstName	Ivan
address	Vitosha 5 Street

- Двойка от ключ и стойност
- Работи само върху браузъри с HTML5
- Може да съхранява информация с размери около 5MB
- Информацията се запазва до затваряне на браузъра или раздела



# Често използвани методи



## **setItem(key, value)**

Добавя или заменя дадения ключ със зададената стойност.

## **getItem(key)**

Връща стойността на даден ключ. Ако този ключ не съществува, връща null.

## **removeItem(key)**

Премахва елемент съдържащ се в sessionStorage с дадения ключ.

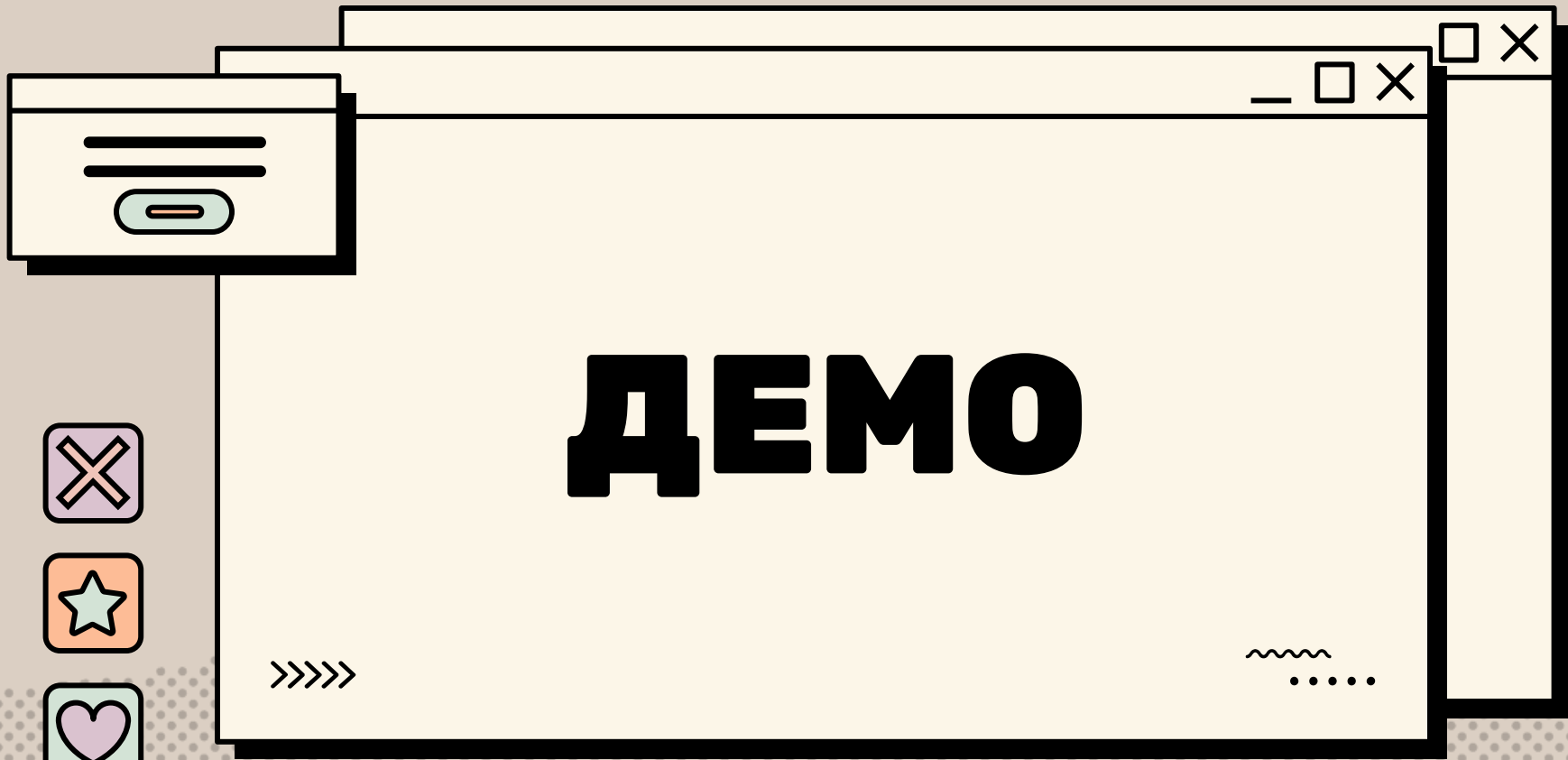
## **clear()**

Премахва цялата информация от sessionStorage



# Пример

```
function onClickNextUser(user) {  
  const userJSON = JSON.stringify(user);  
  sessionStorage.setItem("user", userJSON);  
}  
  
function onClickNextDetail(detail) {  
  const detailJSON = JSON.stringify(detail);  
  sessionStorage.setItem("detail", detailJSON);  
}  
  
function onClickSubmit() {  
  const userJSON = sessionStorage.getItem("user");  
  const detailJSON = sessionStorage.getItem("detail");  
  
  const data = {  
    user: JSON.parse(userJSON),  
    detail: JSON.parse(detailJSON)  
  };  
  
  // Sending the data to a server...  
}
```



ДЕМО





# Кое кога да използваме



## Размер на данните

**Малки**  
Бисквитки

**Големи**  
LocalStorage,  
SessionStorage

## Видове данни

**Постоянни**  
Бисквитки,  
LocalStorage

**Сесийни**  
Бисквитки,  
SessionStorage

## Изолираност на данните

**Изолирани**  
LocalStorage,  
SessionStorage

**Между домейни**  
Бисквитки



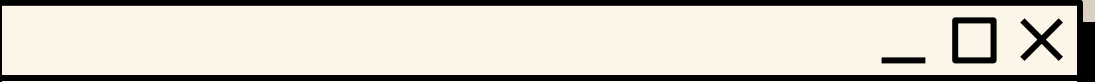
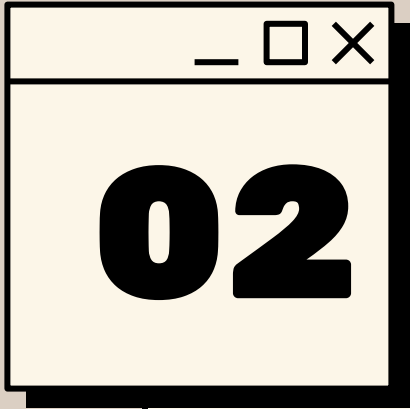
# Добри практики

## Сигурност на данните

- Избягване на съхраняване на чувствителни данни
- При възможност да се криптират на данните
- Използване на HttpOnly атрибута, когато е приложимо.

## Ограничение на размера

- Съхраняване само на най-важните данни
- Информацията, която се пази в бисквитките трябва да бъде минимална, за да се ограничи товара върху мрежата



.....

>>>>

~~~~~



**Cache**

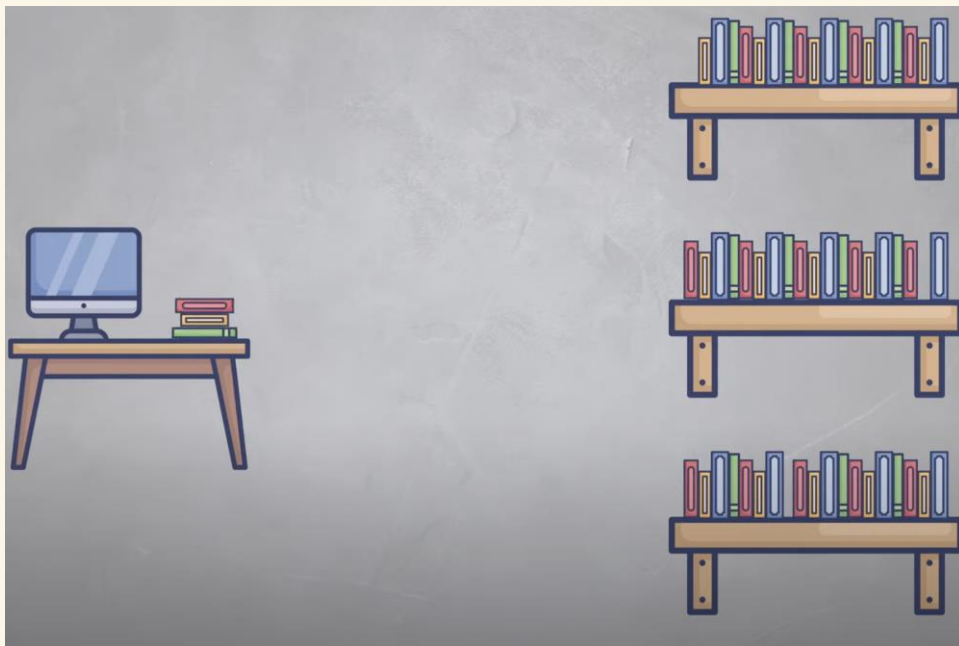




# Какво е Cache?



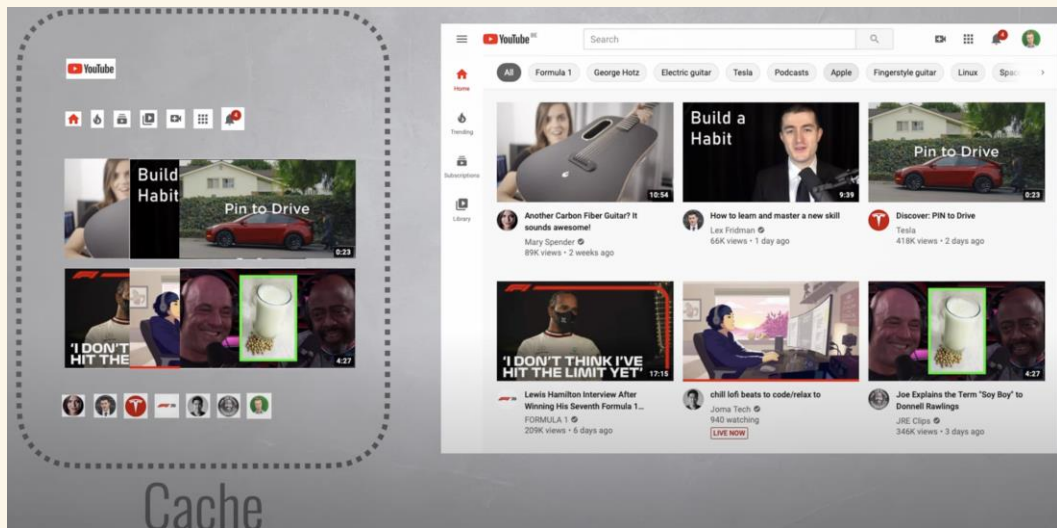
# Какво е Cache?



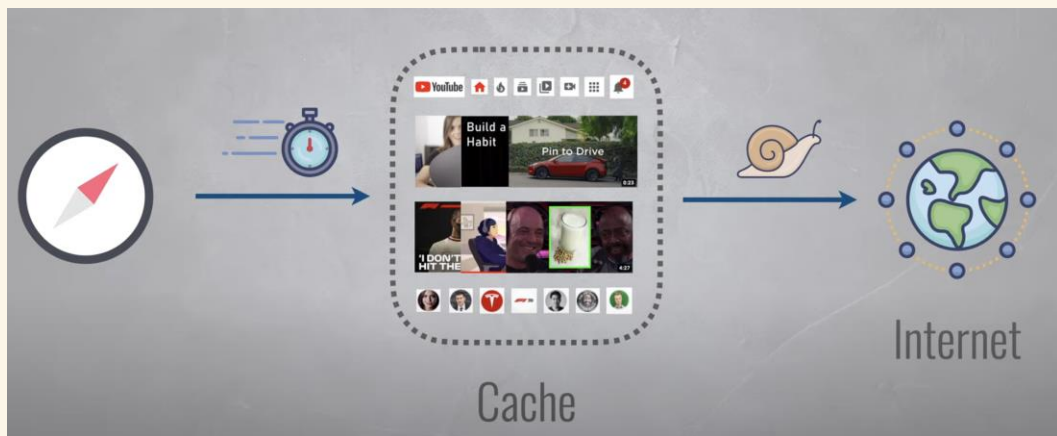
# Какво е Cache?



# Какво е Cache?

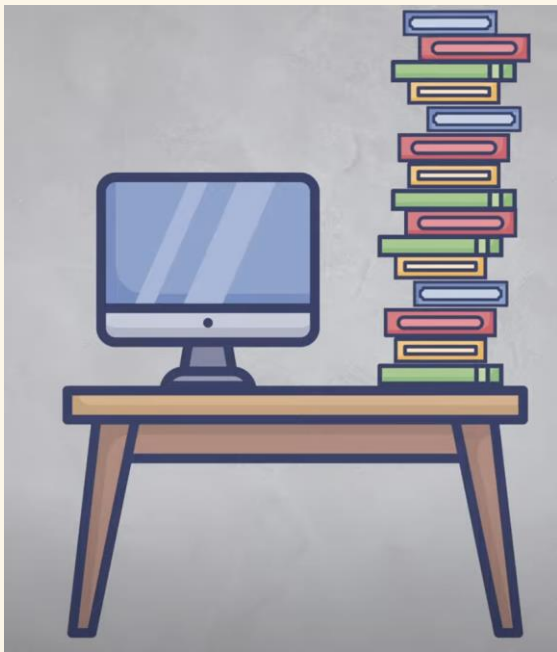


# Какво е Cache?





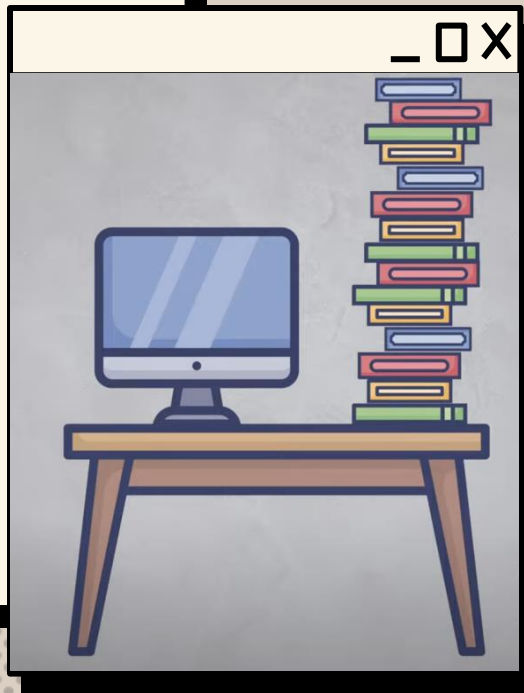
# Какво се прави при пълен кеш?

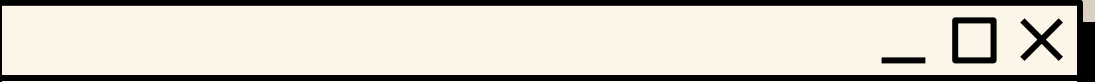
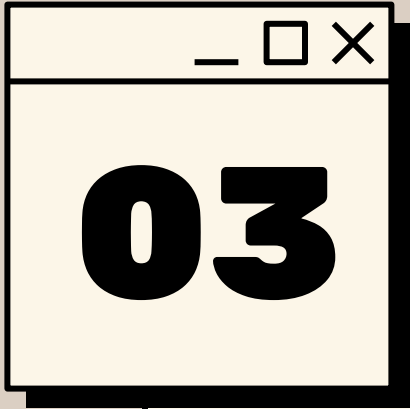


# Методи при пълен кеш

- Least Recently Used (LRU)
- Random replacement

.....



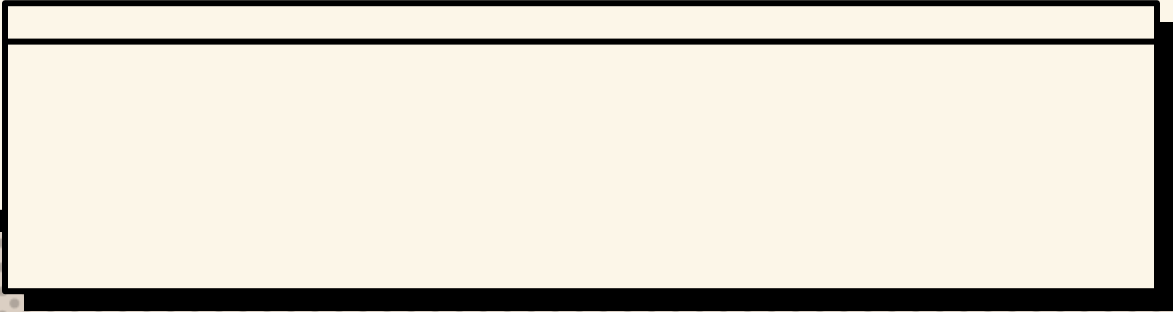


.....

>>>>



# WebSQL





# Какво е WebSQL?

- Web Storage API (Уеб съхраняване)



# Какво е WebSQL?

- Web Storage API (Уеб съхраняване)
- WebSQL е уеб браузър API за съхранение и управление на данни



# Какво е WebSQL?

- Web Storage API (Уеб съхраняване)
- WebSQL е уеб браузър API за съхранение и управление на данни
- За какво се е използвал?



..... **Основни методи на WebSQL** >>>>

## **openDatabase()**

- Създава обект от тип база данни
- Може да създаде нова база данни

## **transaction()**

- Набор от операции в WebSQL

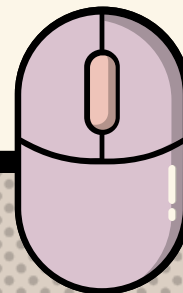
## **executeSQL()**

- Изпълнява една SQL заявка в рамките на трансакция

## Създаване/отваряне на бази данни

```
var dbObj = openDatabase('EmployeeDB', '1.0',  
..... 'Employee Database', 2 * 1024 * 1024);  
.....
```

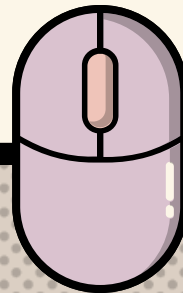
.....





## Създаване на нова трансакция

```
<script>
var dbObj = openDatabase('EmployeeDB', '1.0',
    'Employee Database', 2 * 1024 * 1024);
dbObj.transaction(function (tx) {
    tx.executeSql('CREATE TABLE IF IT DOESNT EXIST Employee_Table (id unique, Name, Age, .)');
});
```

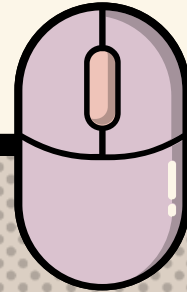


# Вмъкване на нови елементи

```
<script>
var dbObj = openDatabase('EmployeeDB', '1.0',
    'Employee Database', -2 * 1024 * 1024);
dbObj.transaction(function (tx) {
    tx.executeSql('CREATE TABLE IF IT DOESNT EXIST Employee_Table (id unique, Name, Age, )');
});

function Insert(){
    var id = document.getElementById('id').value;
    var name = document.getElementById('name').value;
    var age = document.getElementById('age').value;
    dbObj.transaction(function (tx) {
        tx.executeSql('INSERT INTO Employee_Table (id, Name, Age) values (' + id + ', "' + name + '", ' + age + ')');
    });
}
```

.....





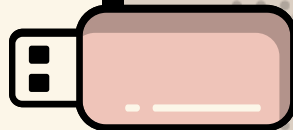
# Съществува ли още WebSQL?

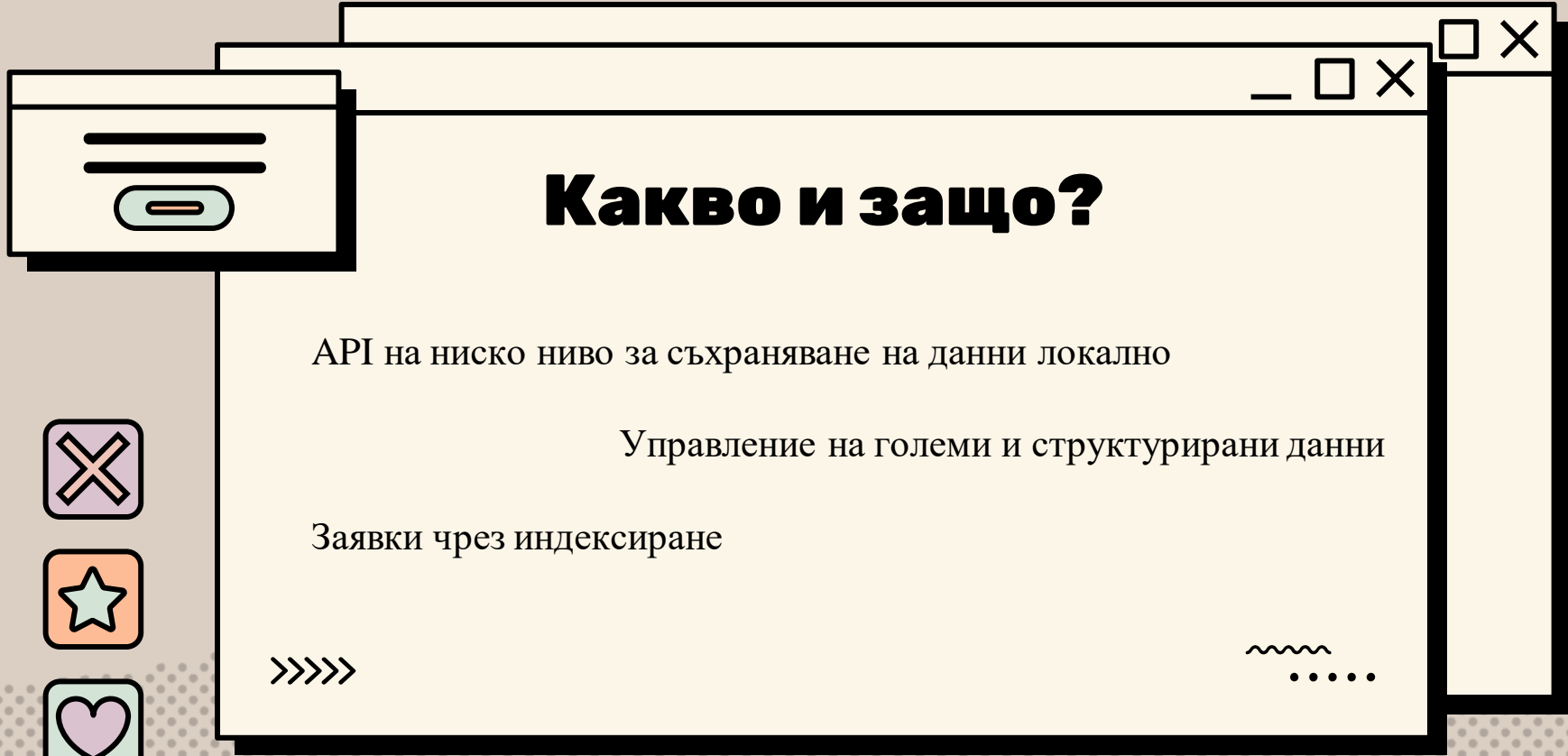


04

# IndexedDB

Преминаване отвъд границите





# Какво и защо?

API на ниско ниво за съхраняване на данни локално

Управление на големи и структурирани данни

Заявки чрез индексирание





# Основни функционалности



## Асинхронност

Предотвратява блокирането на главната нишка

## Индексиране

Бързо и ефективно извличане на данни

## Трансакции

Всички операции се случват чрез трансакции

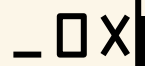


## Офлайн веб приложения



- Безпроблемна работа офлайн
- Съхранява нужните данни локално

## Кеширане



- Отличен механизъм за кеширане
- Подобрява производителността
- Намаляване на заявките към сървъра



## Приложения с интензивни данни



- Работа с големи обеми структурирани данни



..... **Често използвани методи** >>>>

**open(name, version)**

Отваря базата данни

**put(object, [key])**

**add(object, [key])**

Добавя или редактира запис. add хвърля грешка ако вече съществува такъв

**delete(query)**

Премахва запис чрез заявка

**getAll(query)**

Търси записи чрез заявка или индекс





# Пример

```
// Отваряне на базата данни
let request = indexedDB.open("myDatabase", 1);

// Събитие в случай на грешка
request.onerror = function(event) {
    console.log("Database error: " + event.target.errorCode);
};

// Събитие в случай на успех
request.onsuccess = function(event) {
    console.log("Database opened successfully");
};

// Дефиниране на структурата на базата данни при инициализация или актуализация
request.onupgradeneeded = function(event) {
    let db = event.target.result;
    let objectStore = db.createObjectStore("customers", { keyPath: "id" });
    console.log("Defined structure")
};
```



# Пример

```
let db = request.result;
let transaction = db.transaction(["customers"], "readwrite")
let customers = transaction.objectStore("customers")

let customer = {
  id: 1337,
  first_name: "John",
  last_name: "Doe"
}

let addRequest = customers.add(customer)

addRequest.onsuccess = function() {
  console.log("Customer added", addRequest.result);
};

addRequest.onerror = function() {
  console.log("Error", addRequest.error);
};
```



# Пример

```
let db = request.result;  
let transaction = db.transaction(["customers"], "readwrite")  
let customers = transaction.objectStore("customers")
```

```
let customer = {
```

| Key  | Value                                             |
|------|---------------------------------------------------|
| 1337 | {"id":1337,"first_name":"John","last_name":"Doe"} |

```
}
```

```
let addRequest = customers.add(customer)
```

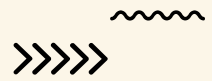
```
addRequest.onsuccess = function() {  
  console.log("Customer added", addRequest.result);  
};
```

```
addRequest.onerror = function() {  
  console.log("Error", addRequest.error);  
};
```



.....

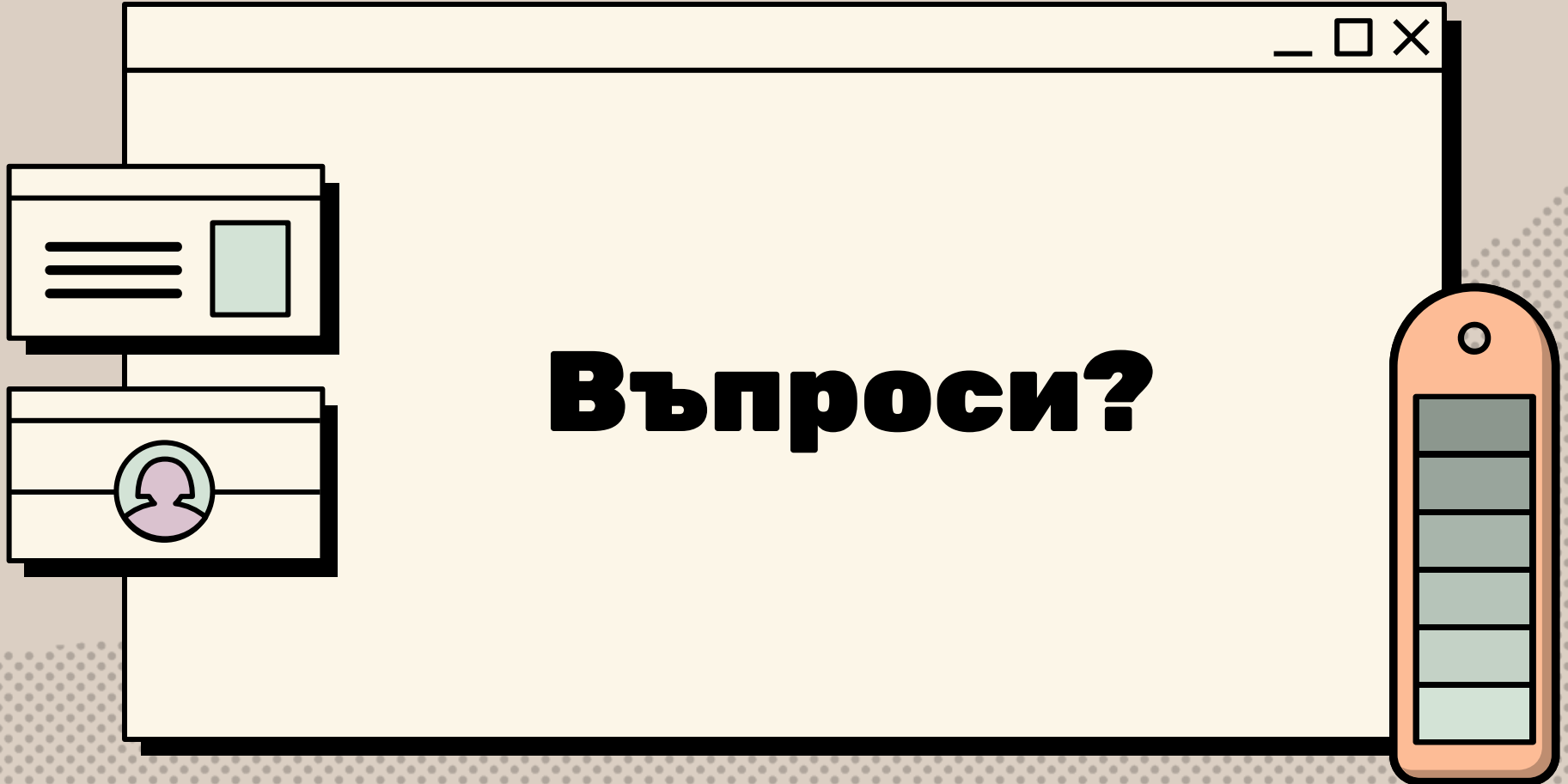
# Обобщение



|                                 | Бисквитки                   | LocalStorage    | SessionStorage         | IndexedDB       |
|---------------------------------|-----------------------------|-----------------|------------------------|-----------------|
| <b>Капацитет</b>                | 4KB                         | ~10MB           | ~5MB                   | $\infty^*$      |
| <b>Достъп</b>                   | Всеки прозорец              | Всеки прозорец  | Даден прозорец         | Всеки прозорец  |
| <b>Изтича</b>                   | На определено зададена дата | Никога          | След затваряне на таба | Никога          |
| <b>Съхранение</b>               | В браузъра и сървъра        | Само в браузъра | Само в браузъра        | Само в браузъра |
| <b>Изпраща се с HTTP заявки</b> | Да                          | Не              | Не                     | Не              |



**Благодарим за  
вниманието!**



**Въпроси?**