-aksesing propertis 1 sa dot.notation primer> console.log(person.name); objekat person -name je propertis..print out Boban

-aksesing propertis 2 sa uglastim zagradama> console.log(restaurant['entrees']); restoran objekat entrees je propertis(lokacija gde se nalazi vrednost)--..print out:Chicken Cacciatore'

-aksesing propertis 3> koristimo varijablu da bi smo selektovali key(propertis unutar objekta), radi se kao i kod aksesing propertisa 2(sa uglastim zagradama)

\*primer

let meal = 'none';

let time = 12;

const restaurantSpecials = {

breakfast: 'jaje',

lunch: 'slanina',

none: 'kurac'

};

if (time < 11) {

meal = 'breakfast'; // moramo negde da izjednacimo varijablu i key kako da kada pozovemo varijablu da se odazove key POY.

} else if (time < 5) {

meal = 'lunch';

}

console.log(restaurantSpecials[meal]);

-adding property > hocemo da dodamo odredjeni key unutar objekta to radimo uz pomoc uglastih zagrada ili dot.notation

izgleda ovako: objekat lokacija vrednost

restaurant['appetizers'] = ['Fried Calamari', 'Bruschetta']; res-objekt; appetizers-property/key

restaurant.desserts = ['Homemade Tiramisu', 'Cannoli'];

-edditing postojecih propertija> samo stavimo nove vrednosti to jes assign-ujemo ...ovako:

restaurant['appetizers'] = ['Fried Calamari', 'Bruschetta', 'Caprese Salad']; // ILI

restaurant.desserts = ['Homemade Tiramisu', 'Canolli', 'Cheesecake'];

-funkcija je metod kada imamo key-funkcija par unutar objekta .

primer:

const restaurant = {

name: 'Italian Bistro',

entrees: ['Penne alla Bolognese'],

openRestaurant: () => {

return 'Unlock the door, flip the open sign. We are open for business!';

},

closeRestaurant: () => {

return 'Lock the door, flip the open sign. We are closed.'

}

};

console.log(restaurant.openRestaurant()); // pozivanje metoda vrsi se ovako , u konzoli se printuje nesto tipa: radimo od 15h dodjite tada

console.log(restaurant.closeRestaurant());

-ES6 Metodi> ne treba im = i 2 tacke izgleda ovako:

openRestaurant() {

return 'Unlock the door, flip the open sign. We are open for business!';

},

-The this Keyword I> da bi metod operisao sa vrednostima koje se nalaze unutar objekta i da bi im pristupio unutar funkcije objekta koristimo THIS -this.hasDineInSpecial

-The this Keyword II> mozemo pozajmiti key/propertis nekom drugom objektu ako smo pre toga koristili unutar naseg objekta

primer:

let myObj = {

name: 'Miti',

sayHello() {

return `${this.name} says hello!`;

}

};

let yourObj = {

name: 'Timer'

};

yourObj.sayHello = myObj.sayHello;

// Sets the sayHello method on yourObj to be the sayHello method on yourObj

---SETTER> vezuje properti objekta za vunkciju DA SE poziva kada postoji pokusaj da se PROMENI /MDN SAJT

> The set syntax binds an object property to a function to be called when there is an attempt to set that property./to jest da je menjas VALUES KOJE SU SAVED TO THESE KEYS

primer: restaurant.seatingCapacity = 150; //nacin na koji pozivamo setter metod da se pozove samo u slucaju kada smo nesto "prckali"(kao u kode akademi sa mestima za sedenje) u vezi sa tim propertisom

var language = {

set current(name) {

this.log.push(name);

},

log: []

}

language.current = 'EN';

console.log(language.log); // ['EN']

----GETTER> The get syntax binds an object property to a function that will be called when that property is looked up.

-Getters are used to get the property values inside of an object.We call the getter method the same way we would access a property without a method: restaurant.seatingCapacity = 150; const seats = restaurant.seatingCapacity;

-in this example we set the seatingCapacity to 150, then call the getter method using restaurant.seatingCapacity and save the result to a variable called seats

primer:

var obj = {

log: ['example','test'],

get latest() {

if (this.log.length == 0) return undefined;

return this.log[this.log.length - 1];

}

}

console.log(obj.latest); // "test".

language.current = 'FA';

console.log(language.log); // ['EN', 'FA']