

# Back End-utveckling i PHP

**50 YHP**

**today( )**

Olika generella dataformat som JSON och  
XML

**week++**

Introduktion till PHP



# **GENERELLA DATAFORMAT**

Fredag 26 Januari 2018

# **Gemensam övning:**

## Repetition av klasser och diagram

# webshop-final-front-end-task

- Sidan skall vara responsiv.
- All data skall sparas i JSON filer, bedöm när de ska sparas i localStorage och när vi ska använda sessionStorage.
- Det skall finnas minst 4 produkter per underkategori.
- Huvudmenyn skall innehålla: start, 4 produktkategorier, info, kontakt, kundvagn.
- Startsidan får ni fylla med valfritt innehåll.
- Det skall finnas plats för en undermeny som visar en lista på underkategorier som tillhör huvudmenykategorierna.
- Vi skall även visa ett rutnät med sammanfattning över alla produkter som skall visas, när vi klickar på en huvudkategori så skall den visa alla produkter från alla underkategorier till huvudkategorin.
- Klickar vi på en produkt ska vi komma till en produktsida med mer info.
- Vi skall kunna lägga produkter i en kundvagn.
- Går vi till kundvagnssidan så skall den visa våra produkter i kundvagnen, samt räkna ihop totalsumman, plus 55kr i frakt.
- Innan vi kan beställa produkterna så ska vi få möjligheten att logga in.
- Skapar vi ett konto så skall kunden fylla i: Namn, Adress, email och telefonnummer, samt klicka i om de vill ha nyhetsbrev.
- Skickar vi vår order så skall vi se en "tack" sida. Samt spara ordern i en JSON fil

# webshop-final-front-end-task

- Sidan skall vara responsiv.
- All data skall sparas i JSON filer, bedöm när de ska sparas i **localStorage** och när vi ska använda **sessionStorage**.
- Det skall finnas minst 4 **produkter** per **underkategori**.
- Huvudmenyn skall innehålla: start, 4 **produktkategorier**, info, kontakt, kundvagn.
- Startsidan får ni fylla med valfritt innehåll.
- Det skall finnas plats för en undermeny som visar en lista på **underkategorier** som tillhör **huvudmenykategorierna**.
- Vi skall även visa ett rutnät med sammanfattning över alla **produkter** som skall visas, när vi klickar på en **huvudkategori** så skall den visa alla **produkter** från alla **underkategorier** till **huvudkategorin**.
- Klickar vi på en **produkt** ska vi komma till en produktsida med mer info.
- Vi skall kunna lägga **produkter** i en **kundvagn**.
- Går vi till kundvagnssidan så skall den visa våra **produkter i kundvagnen**, samt räkna ihop totalsumman, plus 55kr i **frakt**.
- Innan vi kan beställa **produkterna** så ska vi få möjligheten att logga in.
- Skapar vi ett konto så skall kunden fylla i: **Namn**, **Adress**, **email** och **telefonnummer**, samt klicka i om de vill ha **nyhetsbrev**.
- Skickar vi vår order så skall vi se en "tack"-sida. Samt spara ordern i en JSON fil

# webshop-final-front-end-task

- Sidan skall vara responsiv.
- All data skall sparas i JSON filer, bedöm när de ska sparas i **localStorage** och när vi ska använda **sessionStorage**.
- Det skall finnas minst 4 **produkter** per **underkategori**.
- Huvudmenyn skall innehålla: start, 4 **produktkategorier**, info, kontakt, kundvagn.
- Startsidan får ni fylla med valfritt innehåll.
- Det skall finnas plats för en undermeny som visar en lista på **underkategorier** som tillhör **huvudmenykategorierna**.
- Vi skall även visa ett rutnät med sammanfattning över alla **produkter** som skall visas, när vi *klickar* på en **huvudkategori** så skall den visa alla **produkter** från alla **underkategorier** till **huvudkategorin**.
- *Klickar vi på en produkt ska vi komma till* en produktsida med mer info.
- Vi skall kunna *lägga produkter i en kundvagn*.
- Går vi till kundvagnssidan så skall den visa våra **produkter i kundvagnen**, samt *räkna ihop totalsumman*, plus 55kr i **frakt**.
- Innan vi kan *beställa produkterna* så ska vi *få möjligheten att logga in*.
- *Skapar vi ett konto* så skall kunden fylla i: **Namn**, **Adress**, **email** och **telefonnummer**, samt klicka i om de vill ha **nyhetsbrev**.
- *Skickar vi vår order* så skall vi se en "tack"-sida. Samt *spara ordern* i en JSON fil

# webshop-final-front-end-task

- Sidan skall vara responsiv.
- All data skall *sparas* i JSON filer, bedöm när de ska sparas i **localStorage** och när vi ska använda **sessionStorage**.
- Det skall finnas minst 4 **produkter** per **underkategori**.
- Huvudmenyn skall innehålla: start, 4 produktkategorier, info, kontakt, kundvagn.
- Startsidan får ni fylla med valfritt innehåll.
- Det skall finnas plats för en undermeny som visar en lista på **underkategorier** som tillhör **huvudmenykategorierna**.
- Vi skall även visa ett rutnät med sammanfattning över alla **produkter** som skall visas, när vi *klickar* på en **huvudkategori** så skall den visa alla **produkter** från alla **underkategorier** till **huvudkategorin**.
- *Klickar* vi på en **produkt** ska vi komma till en produktsida med mer info.
- Vi skall kunna *lägga* **produkter** i en **kundvagn**.
- Går vi till kundvagnssidan så skall den visa våra **produkter** i **kundvagnen**, samt *räkna ihop totalsumman*, plus 55kr i **frakt**.
- Innan vi kan *beställa* **produkterna** så ska vi få *möjligheten att logga in*.
- *Skapar* vi ett konto så skall kunden fylla i: **Namn**, **Adress**, **email** och **telefonnummer**, samt klicka i om de vill ha **nyhetsbrev**.
- *Skickar* vi vår order så skall vi se en "tack"-sida. Samt *spara ordern* i en JSON fil

# Generella dataformat, markup

- Skrivs som ren text (jämfört med bilder, ljud mm, dessa *kallas för binärfiler*)
- Har en bestämd syntax, precis som kod, som beskriver innehållet.
- Har inget bestämt användningsområde.
- **Markup filer GÖR aldrig någonting**
- **Exempel: XML, JSON, YAML, MARKDOWN**
- HTML, CSS?

```
{  
    "firstName": "John",  
    "lastName": "Smith",  
    "isAlive": true,  
    "age": 27,  
    "address": {  
        "streetAddress": "21 2nd Street",  
        "city": "New York",  
        "state": "NY",  
        "postalCode": "10021-3100"  
    },  
    "phoneNumbers": [  
        {  
            "type": "home",  
            "number": "212 555-1234"  
        },  
        {  
            "type": "office",  
            "number": "646 555-4567"  
        },  
        {  
            "type": "mobile",  
            "number": "123 456-7890"  
        }  
    ],  
    "children": [],  
    "spouse": null  
}
```

```
# sequencer protocols for Laser eye surgery
---
- step: &id001                      # defines anchor label &id001
  instrument: Lasik 2000
  pulseEnergy: 5.4
  pulseDuration: 12
  repetition: 1000
  spotSize: 1mm

- step: &id002
  instrument: Lasik 2000
  pulseEnergy: 5.0
  pulseDuration: 10
  repetition: 500
  spotSize: 2mm
- step: *id001                      # refers to the first step
- step: *id002                      # refers to the second step
- step:
  <<: *id001
  spotSize: 2mm                      # redefines just this key
- step: *id002
```

Heading

=====

## Sub-heading

Paragraphs are separated  
by a blank line.

Two spaces at the end of a line  
leave a line break.

Text attributes *italic*,  
**bold**, `monospace`.

Horizontal rule:

---

Bullet list:

- \* apples
- \* oranges
- \* pears

Numbered list:

1. wash
2. rinse
3. repeat

```
<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
      Two of our famous Belgian Waffles with plenty of real maple
      syrup
    </description>
    <calories>650</calories>
  </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>
      Light Belgian waffles covered with strawberries and whipped
      cream
    </description>
    <calories>900</calories>
  </food>
  <food>
    <name>Berry-Berry Belgian Waffles</name>
    <price>$8.95</price>
    <description>
      Light Belgian waffles covered with an assortment of fresh
      berries and whipped cream
    </description>
    <calories>900</calories>
  </food>
</breakfast_menu>
```

# **Generella dataformat, markup**

- **Varför?** För att rena textfiler blir svårlästa.
- För att införa **metadata**
- **Enkelt att läsa av både människor och datorer**
- Ett bra mellanalternativ för mindre mängd information, innan man behöver en databas.

# Generella dataformat, markup

- **Varför inte?** Vid för stor mängd blir det oftast klumpigt (som med stora hemsidor)
- Databaser är snabbare.
- Binärfiler (mp3, jpg etc) är mycket snabbare att ★parsa★ för en dator.

# XML

- Står för eXtensible Markup Language
- Mest kända av SGML:formaten, tillsammans med HTML\*
- Det finns inga förbestämda taggar (som p, div etc)
- Taggar kan definieras i ett XML-schema
- Det finns många färdiga parsers



# XML

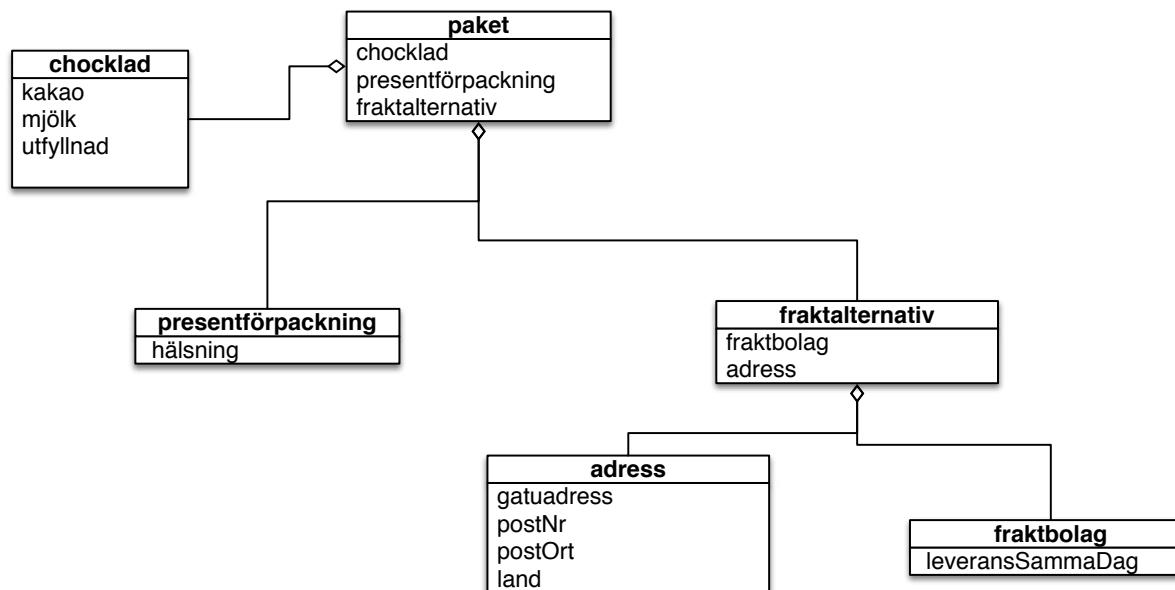
- XML kan ses som ett SGML språk som passar allt, medans HTML "är" ett SGML språk för hemsidor.
- <TAG attribut="värde">innehåll</TAG>
- <SjälvstängandeTag/>
- <Parent><Child></Parent>
- <METADATA>DATA</METADATA>

```
<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
      Two of our famous Belgian Waffles with plenty of real maple
      syrup
    </description>
    <calories>650</calories>
  </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>
      Light Belgian waffles covered with strawberries and whipped
      cream
    </description>
    <calories>900</calories>
  </food>
  <food>
    <name>Berry-Berry Belgian Waffles</name>
    <price>$8.95</price>
    <description>
      Light Belgian waffles covered with an assortment of fresh
      berries and whipped cream
    </description>
    <calories>900</calories>
  </food>
</breakfast_menu>
```

# Övning: XML "Databas"

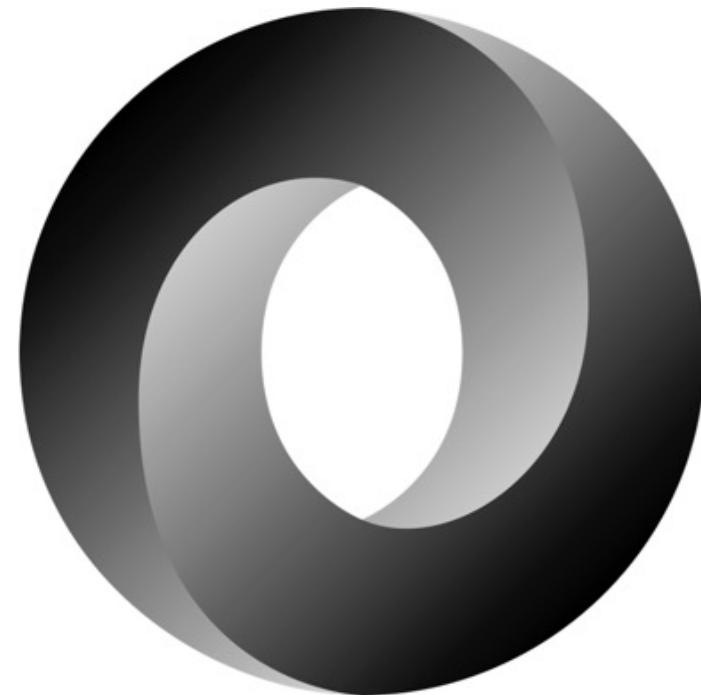
Anton Berg tillverkar choklad med olika mängd kakao, mjölk och utfyllnad.

Nu vill dom skapa en hemsida där man kan beställa egen **choklad** som skickas direkt hem till en **adress**, i valfri **presentförpackning** och med en hälsning som man ska kunna skriva in på hemsidan. Man ska även kunna välja att få **paketet** levererat samma dag med ett dyrare **fraktalternativ**.



# JSON

- **JavaScript Object Notation.**
- Uppfanns för att XML  
innehåller för mycket onödig  
information för att skicka fram  
och tillbaka mellan en server  
och klient
- Nästan likadan syntax som JS  
objekt, men JSON är inte  
bundet till JavaScript
- Parser inbygd i Javascript



# JSON

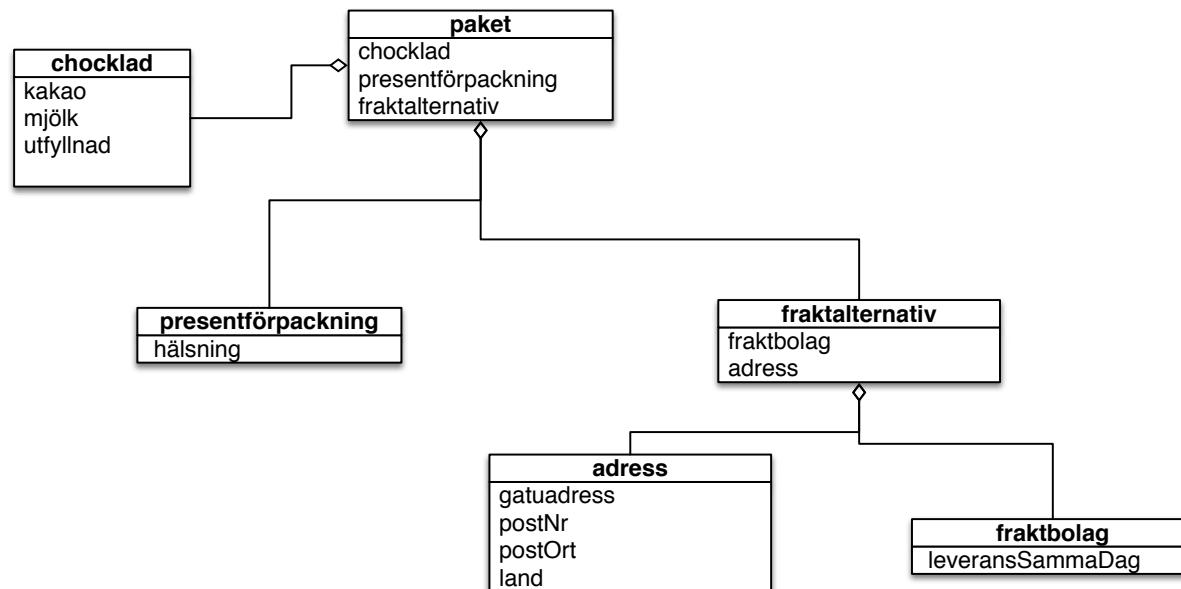
- JSON består av ett { key: value... } objekt.
- Key är alltid en sträng.
- Value är en sträng, lista, number, boolean, ett objekt eller *null*
- { Key: Value }
- { METADATA: DATA }

```
{  
    "firstName": "John",  
    "lastName": "Smith",  
    "isAlive": true,  
    "age": 27,  
    "address": {  
        "streetAddress": "21 2nd Street",  
        "city": "New York",  
        "state": "NY",  
        "postalCode": "10021-3100"  
    },  
    "phoneNumbers": [  
        {  
            "type": "home",  
            "number": "212 555-1234"  
        },  
        {  
            "type": "office",  
            "number": "646 555-4567"  
        },  
        {  
            "type": "mobile",  
            "number": "123 456-7890"  
        }  
    ],  
    "children": [],  
    "spouse": null  
}
```

# Övning: skapa en JSON "Databas"

Anton Berg tillverkar choklad med olika mängd kakao, mjölk och utfyllnad.

Nu vill dom skapa en hemsida där man kan beställa egen **choklad** som skickas direkt hem till en **adress**, i valfri **presentförpackning** och med en hälsning som man ska kunna skriva in på hemsidan. Man ska även kunna välja att få **paketet** levererat samma dag med ett dyrare **fraktalternativ**.



one more thing...

Wat

@garybernhardt

Trevlig Helg!