

# Testautomatisering

HTML, Dev Tools, Watir

- Schemaändring
  - Samma tider som innan
  - Mån, FM: Intro-lektion, veckans topic
  - Fre, FM: Liknande verktyg, veckans topic + Video
    - Kan strykas till förmån för extra handledning
    - Vi reder ut det med handuppräckning i början av lektionen
    - Slajds + Länk till video kommer finnas för de som ej behöver handledning.

**Korta punkter**

- FM:
  - HTML Crash Course
  - Browser Dev Tools
  - Watir (och Record/Replay vs. WebDriver)
- EM:
  - Virtual Machine Intro
  - Labbintro

**Idag**

- Watir är inte Record/Replay á la Selenium
- Identifiera intressanta element
- Kunskap om HTML
- Dev Tools

**Varför HTML + Dev Tools?**

- Vad är HTML?
- Hierarkiskt ordnad information på en hemsida
- 
- Vad är en webb-sida?
  - HTML (Hierarki + information)
  - CSS (layout / utseende)
  - javascript (beteende)

# Crash Course HTML

- Enkelt exempel:

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Sidtitel i tabbarna</title>
</head>
<body>
  <a href="http://www.google.com">Google</a>
</body>
</html>
```

# Crash Course HTML

- Tag / HTML Element – Grundbulten i HTML
- inom "<" och ">"
- Normalt starttag och slutttag. Slutttag markeras med "/"
  - ex: "<html></html>"
  - "<h1>Jag är en rubrik</h1>"
  - i vissa fall ingen slutttag: "<img>", "<input>"
  - i vissa fall inbakad slutttag: "<input />"

# Crash Course HTML

- Namnet på taggen - säger vad taggen är till för
- ex:
  - "<input>" - någon form av inmatningsfält
  - "<img>" - en bild
  - "<p>" - en paragraf
  - "<button>" - en knapp
- Det finns en uppsjö av taggar – de flesta är dock inte särskilt intressanta för oss.

# Crash Course HTML



- Utöver tag-namnet så har en tag ytterligare information
- Nästan alla denna information är valfri
- id - en unik identifikation för taggen
  - ex: "<img id="IMG\_logo">"
  - Om vi hittar en tagg med id="IMG\_logo" så är det bild-taggen ovan.

# Crash Course HTML

- class – anger vilken CSS-klass taggen har. Men vi kan också använda class för att identifiera en viss tagg
- Om det bara finns en knapp "<button>" med klassen "class='SaveButton'" så kan vi identifiera knappen.
- Fråga efter ett element "button" med klassen "SaveButton"  
- trots att knappen kanske saknar id
- Ex: "<button class='SaveButton'>"

## Crash Course HTML

- attributes - ytterligare bestämning för taggen. Olika taggar har olika attribut
- ex:
  - "<input type='checkbox'>"
  - "<img src='logo.jpg'>"

# Crash Course HTML

- Ett antal exempel på vanliga taggar

# Crash Course HTML

- gruppering: "<div>"
- Ett antal element som på något sätt hör samman
- ex:
  - "<div id="menu">  
...Ytterligare element..."
  - "</div>"

# Crash Course HTML

- text: normalt innanför "<p>", "<h1>", "<span>", "<div>"-taggar
- men många andra taggar kan innehålla text - t.ex. "<body>Denna text kommer visas</body>"
- I stort sett alla taggar som består av en start-tagg och en slut-tagga kan innehålla text.

## Crash Course HTML

- inmatningsfält: "<input>"
  - ex: "<input type='text'>"
  - ex: "<input type='radio'>"
  - ex: "<input type='checkbox' checked='true'>"
  - ex: "<input type='password'>"
  - ex: "<input type='button' value='Save Button'>"
  - ex: "<input type='submit'>"
  - ex: "<input type='hidden'>"
  - ex: "<input type='image'>"
- Type image och button kommer behandlas som en button av watir.

# Crash Course HTML

- Inmatningsfält, Nyare, ej lika vanliga ännu
  - ex: "<input type='date'>"
  - ex: "<input type='email'>"
- etc... finns totalt 24 st. De 8 första jag visade är vanligast

# Crash Course HTML



- Länkar: "<a>" (anchor-tag)
  - ex: "<a href='\"http://www.google.com>Google</a>\""
- ...länkar kan också vara en omstylad knapp, eller t.ex. en "<div>"

# Crash Course HTML

- Knappar: "<button>", "<input type='button'>",
- Men: kan också vara en omstylad "<a>"-tag
- Men: kan också vara en omstylad "<div>"-tag
- Nästan alla taggar går att styla om – många taggar går att göra så de ser ut som knappar.
- Om ni räknar med att alla element kan göras klickbara så är ni inte helt fel ute
  - (Utvecklaren har använt javascript för att ändra standardbeteendet för element).

## Crash Course HTML

- Form: "<form>...</form>"
- formulär.
- Data som kommer skickas tillsammans till servern. Ex. "Fyll i dina personliga uppgifter"
- Kommer för det mesta med en "<input type='submit'>"

# Crash Course HTML

- Break: `<br />`
  - Line Break
  - Bör eg. inte användas

# Crash Course HTML

- Exempel på HTML-element
- Jag sitter i en vanlig text-editor.
- Jag laddar html-filen direkt i browsern

# Crash Course HTML

- Exempel på HTML-element
- Jag sitter i en vanlig text-editor.
- Jag laddar html-filen direkt i browsern
- Vi kan se hur html för en sida ser ut genom högerklick -> Visa källa.

# Crash Course HTML

- Varning:
  - HTML är i praktiken väldigt ofta en sörja
  - id behöver inte vara unikt
    - Problem?
    - Lösningar?
  - HTML används inte enligt standard
    - Problem?
    - Lösningar?
  - Djup och komplex hierarki
    - Problem?
    - Lösningar?

# Crash Course HTML

- Resurser:

- <https://developer.mozilla.org/en-US/docs/HTML>

- Introduction to HTML

- HTML element reference

- Missa inte: "See also"

- HTML Attributes list

- Glömt hur en checkbox taggas? Googla "HTML Checkbox" (Engelska sökningar bättre än svenska)

# Crash Course HTML



- Varför?
  - Hitta element
  - Se fel på sidan (Avsaknad av resurser, Javascript-fel, etc)
  - Se hur lång tid sidan tar att ladda
  - Se storleken på sidan
  - mm...
    - (för utvecklare: t.ex. debugga javascript)
    - (för designers: t.ex. ändra färger, flytta element on the fly)

# Crash Course Developer Tools

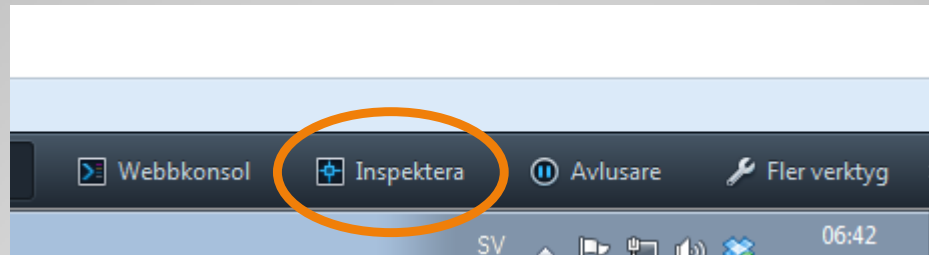
- Varför?
  - Låt er inte luras av namnet - kan vara ett utmärkt hjälpmedel för er

**Crash Course Developer Tools**

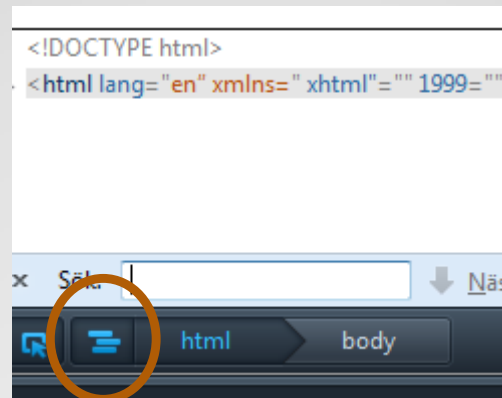
- Vi kommer bara kika på "inspect element" idag.

**Crash Course Developer Tools**

- Firefox:
- Firefox -> Webbutvecklare -> Utvecklarfältet
- Inspektera Element

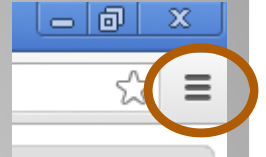
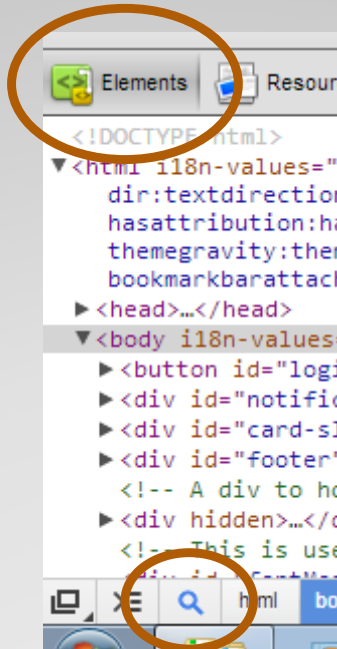


- Märkpanel



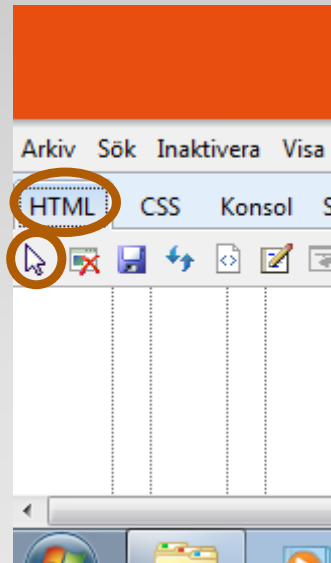
# Crash Course Developer Tools

- Chrome:
- Meny -> Verktyg -> Verktyg för programmerare
- Select...



# Crash Course Developer Tools

- IE:
- Verktyg -> F12-Utvecklingsverktyg
- Välj...



# Crash Course Developer Tools

- Watir

**Watir**

- Watir:
  - Vad?
    - Scripta End-to-end test i browser m. ruby
    - Samma typ av tester som Record/Replay-tester

**Watir**



- Stödjer:
  - IE (Endast Windows)
  - Chrome
  - FF
  - Safari (Inte helt 100)
  - Opera
  - Etc.
  - Vi kan även göra test på Android/iPhone-lurar

**Watir**

- Vilka använder Watir?



**Watir**

- Alternativ
  - WatiN (C#/.Net)
  - WatiJ (Java)
  - Selenium IDE / Selenium WebDriver
  - => Watir är multiplatform/multispråk

**Watir**

- Fullt automatiserade test
  - Ex: Ett automatiserat End-to-end test
- Halvt automatiserade test
  - Ex: Hjälp med att fylla i formulär vid manuella tester

**Watir**

- Selenium vs. Watir

**Watir**

- Selenium vs. Watir
  - Watir och selenium WebDriver delar grundkomponenter

**Watir**

- Selenium vs. Watir

- Record/Replay (Selenium IDE) ej möjligt med watir out of the box
- Record/Replay går snabbare att utveckla och är väldigt mycket enklare att lära sig
  - => Men! När man suttit med watir ett tag så är skillnaden inte särskilt stor
  - => Men! Merparten av tiden för att utveckla tester av den här typen är inte implementation

**Watir**

- Selenium vs. Watir
  - Scriptade test (WebDriver)
  - Många tycker att watir är enklare att lära sig

**Watir**



- Selenium vs. Watir
  - Watir är kraftfullare out of the box

**Watir**

- Selenium vs. Watir
  - Varför Watir i denna kurs?
  - En anledning att kika på Watir: Exponering för fler verktyg

**Watir**

- Selenium vs. Watir
  - Varför Watir i denna kurs?
  - En bättre anledning: Record/Replay har vissa problem jämfört med WebDriver-test.

**Watir**

- Problematik med Record/Replay
- Record/Replay-tester generellt sett mer instabila och går lättare sönder vid uppdateringar av SUT

- Problematik med Record/Replay
- Record/Replay-tester spelar in onödiga saker (detta bidrar till instabiliteten)

- Problematik med Record/Replay
- Vissa element är svårare att spela in: T.ex: Frames, popups

- Problematik med Record/Replay
- Selenium har historiskt sett haft större timing-problem (bidrar till instabiliteten)

- Problematik med Record/Replay
- Vi kan inte köra record/replay-tester på en build server/CI server hur som helst
  - => Exempel/målbild nästa sida



Jenkins

ENABLE AUTO REFRESH

add description

New Job

People

Build History

Edit View

Project Relationship

Check File Fingerprint

Manage Jenkins

Disk usage

## Build Queue








































No builds in the queue.



## Build Executor Status

#	Status
1	Idle
2	Idle

## Next executions

	12/05/2012 00:00
	12/05/2012 01:30
sonar build	12/05/2012 02:30
sonar build	12/05/2012 03:30

All	All Jobs	Disabled jobs	branch jobs	problematic jobs	radiator	+	
S	W	Name ↓	Last Statuses	Test Result	Next Launch	Cron Trigger	
		<a href="#">[redacted]</a>	4.1 min > 1.8 hr > 2.1 hr	(no failures)	 12/05/2012 00:00	Build periodically: @midnight Poll SCM: *****	
		<a href="#">[redacted]</a>	1.3 mo	N/A			
		<a href="#">[redacted]</a> Deploy Test Environment	1.2 days > 3.6 days	N/A	 12/05/2012 01:30	Build periodically: 30 1 *** 0 11 ***	
		<a href="#">[redacted]</a>	2.9 days > 4.3 days	(no failures)		Poll SCM: *****	
		<a href="#">[redacted]</a> sonar build	1.5 days > 3.5 days	(no failures)	 12/05/2012 03:30	Build periodically: 30 3 ***	
		<a href="#">[redacted]</a>	1.2 hr > 3.4 hr > 3.9 hr	(no failures)		Poll SCM: *****	
		<a href="#">[redacted]</a> GUI Tests	2.2 hr > 3.6 hr	(no failures)			
		<a href="#">[redacted]</a> Integration Tests	39 min > 3.6 hr > 1 day	(no failures)		Poll SCM: @hourly	
		<a href="#">[redacted]</a> sonar build	1.5 days	(no failures)	 12/05/2012 02:30	Build periodically: 30 2 ***	
		<a href="#">[redacted]</a> Test Deploy	1.3 mo > 2.3 mo	N/A			

Icon: [S](#) [M](#) [L](#)Legend  RSS for all  RSS for failures  RSS for just latest builds

Watir

- CI (Jenkins)
- Översiktspanel - vi kan även generera mails, plugins för att exportera till andra format/program

**Watir**

- Problematik med Record/Replay
- Vi kan inte integrera record/replay-tester med Unit Tests/BDD-tester hur som helst
  - => Vi kommer kika på detta under senare veckor i kursen.

- Både selenium och watir utvecklas aktivt
  - ~3-10 releases/år
- Speciellt WebDriver är fortfarande ganska ungt

**Watir**

- Vad skall jag välja?
  - Några scenarion

**Watir**

- Scenario 1: Vi behöver snabbt få upp automatiserade end-to-end test för några vanliga uppgifter.
  - t.ex. tidig utveckling - vi har inte nått upp till en stabil produkt ännu och itererar snabbt
- => Kör record/replay

**Watir**

- Scenario 2: Det är inte viktigt att dessa tester fungerar om 6 månader
  - t.ex. en kampanjsite som skall ligga upp 1 månad
- => Kör record/replay

- Scenario 3: Det du försöker testa är inte möjligt att spela in
- Scenario 4: Ett långsiktigt perspektiv - underhåll av dessa test är viktiga
- Scenario 5: Testerna skall köras på en Build Server (CI)
  - "normalfallet"?
- => WebDriver

**Watir**



- Det sagt: När ni kommer ut i arbetslivet...
  - => Om ni känner er bekväma med Record/Replay - kör på det
  - => Om ni har utrymme att experimentera - vidga era kunskaper till sådant ni är mindre bekväma med
  - => Om ni stöter på patrull med Record/Replay - håll i bakhuvudet att scripting med ex. watir är en möjlighet som kan lösa era problem

**Watir**

- Både selenium och watir utvecklas aktivt
  - ~3-10 releases/år
- Speciellt WebDriver är fortfarande ganska ungt

**Watir**

- Att använda Watir:
  - definiera vilken browser vi vill använda
    - `browser = Watir::Browser.new :ie`

**Watir**

- Att använda Watir:
  - gå till en URL
    - `browser.goto 'bit.ly/watir-webdriver-demo'`

**Watir**

- Att använda Watir:
  - verifiera att någonting finns på sidan vi är på
    - `div = browser.div :class => 'ss-form-desc ss-no-ignore-whitespace'`
    - `div.exists?`

**Watir**

- Att använda Watir:
  - vilken typ av element vill vi göra någonting med?
    - HTML element / taggar
  - hur identifierar vi det specifika element vi vill göra någonting med?
    - Vi kommer långt med id, name – mer på fredag
    - Även XPath – mer på fredag
  - Vad vill vi göra med detta element?
    - Nu: .click

**Watir**

- Att använda Watir:
  - Mer komplex interaktion: Fredag

**Watir**

- vänta på X innan vi utför nästa sak
  - Vänta på att en viss tid passerat
  - Vänta på att ett villkor uppfylls
  - Vänta på att ett objekt försvinner
  - Vänta på att ett objekt dyker upp
- (resurs för vänta: <http://watirwebdriver.com/waiting/>)

**Watir**



- vänta på X innan vi utför nästa sak

```
require 'watir-webdriver'
```

```
b = Watir::Browser.start 'bit.ly/watir-webdriver-demo'
```

```
b.button(:value => 'Submit').wait_while_present
```

```
b.select_list(:id => 'entry_1').wait_until_present
```

```
b.text_field(:id => 'entry_0').when_present.set 'your name'
```

```
Watir::Wait.until { b.text.include? 'Thank you' }
```

**Watir**

- Resurser - Watir:

- <http://watirwebdriver.com/>
- <http://wiki.openqa.org/display/WTR/Tutorial>
- Watir-boken: <http://watir.github.com/watirbook/> (Anses vara 30% färdig)
- <http://watir.com/examples/> (Snabba exempel)

**Watir**

- Några allmänna ord om verktygen
  - De flesta verktyg i den här klassen
  - ...förutsätter en viss datorvana
  - ...har inte någon fokus på användbarhet
- => Räkna inte med tydliga felmeddelanden

## Installation VBox

- ...kryptiska felmeddelanden i textlog - för det mesta hjälper googling
  - ...var uppmärksam på fel
- => För guider: Läs, Läs igen - övertyga dig själv om att du inte missat ngt steg. Fundera även ifall du gjort antaganden som kanske inte stämmer

## Installation VBox

- Vid fel
- Min erfarenhet:
  - Ngt steg i installationsförfarandet missades
  - => felmeddelande har i senare skede visats, men missats

# Installation VBox

- Några ord om miljö:
  - Om mig
  - => Min standardmiljö: Windows
  - => Jag tycker om Windowsmiljön
  - => Dvs. Jag är inte en linux-fanatiker

## Installation VBox

- När ni jobbar med de flesta scriptspråk (ruby, php, python, etc.)
  - => Föredra \*nix (Linux/Unix/BSD)

## Installation VBox

- När ni jobbar med de flesta scriptspråk (ruby, php, python, etc.)
  - => Information
  - => ofta utvecklat för \*nix

## Installation VBox



- När ni jobbar med de flesta scriptspråk (ruby, php, python, etc.)
  - => kontentan: \*nix är generellt sätt mer friktionsfritt för språk och verktyg i denna kategori
  - => Hade vi jobbat i WatiN (C#/.Net-variant av Watir) så hade Windows varit miljön att föredra

## Installation VBox

- Snabbintro - xubuntu + sublime text + ruby + watir-webdriver

## Installation VBox

- Målsättning med labbar:
  - Hands on!
  - Visa grundläggande, men nyttiga scenarion/tekniker
- => Jag hoppas att ni efter labbarna skall se att ni har någon nytta av de tekniker vi utnyttjar

## Installation VBox

- Målsättning denna lab:
  - Mjukstart
  - => Jag räknar med vissa installationsproblem
  - => Schemaändring
  - => Om ni inte uppfattar det som mjukstart: Säg till

# Installation VBox

- Om otydligt formulerad fråga: Säg till direkt.

# Installation VBox

- Handledning:
- För "G"-nivå: Väldigt hjälpsam (Räkna med mer hints på tisdagen och mer konkreta saker på fredagen dock)
  - - Räkna med att söka "viss" information på egen hand
- För "VG"-nivå: Jag kommer hjälpa er, men jag kommer vara mycket mindre konkret
  - - räkna med att ni behöver söka en hel del information på egen hand här

## Installation VBox

- Om ni har problem: Gör minst en googling i syfte att lösa problemet
- Varför vill jag att ni skall söka information på egen hand?
  - Hela hemligheten med att bli IT-proffs: Bli bra på att Googla och snabbt utvärdera ifall informationen ni hittat är vettig

## Installation VBox

- Feedback

- Tänk inte att någon annan säger till ifall det är för svårt
- Om endast en elev säger till mig => mindre troligt att jag ändrar något

# Installation VBox



- Ge inte upp för lätt!
  - Kan du bryta ned problemet i mindre delar
    - - är det något av de mindre problemen du kan lösa?
  - Har du gjort något liknande som du kan transformera till en lösning på problemet?
  - Kan du börja i någon annan ände?
  - Experimentera!

# Installation VBox

- Google + Självförtroende + Pannben = Lösta problem

# Installation VBox

- Inga överkursuppgifter

- => Testa CI-tjänst/Sätt upp CI-server/Build-server (ex: Jenkins)
- => Scripta upp en Testmiljö (ex: Vagrant)
- => Utforska själv veckans ämne djupare
- => Utforska programmering vidare (ex: webbutveckling)
- => Utforska något av de ämnen vi tar upp på fredagar
- => Utforska något tidigare ämne ni studerat djupare
- => Jag bollar gärna idéer med er
- => Jag kommer fokusera på de som har frågor kring labbarna, så räkna med att vara hyfsat självgående

# Installation VBox

- Watir – fördjupning
  - Mer avancerade scenarion
  - XPath – om det är omöjligt att identifiera ett element
  - Exempel – maila gärna
- Whitebox vs. Blackbox testing
  - Har ni pratat om detta innan?
- Scripts för enklare test
  - T.ex. för att testa ett Web API

**Nästa gång**

...

Fin