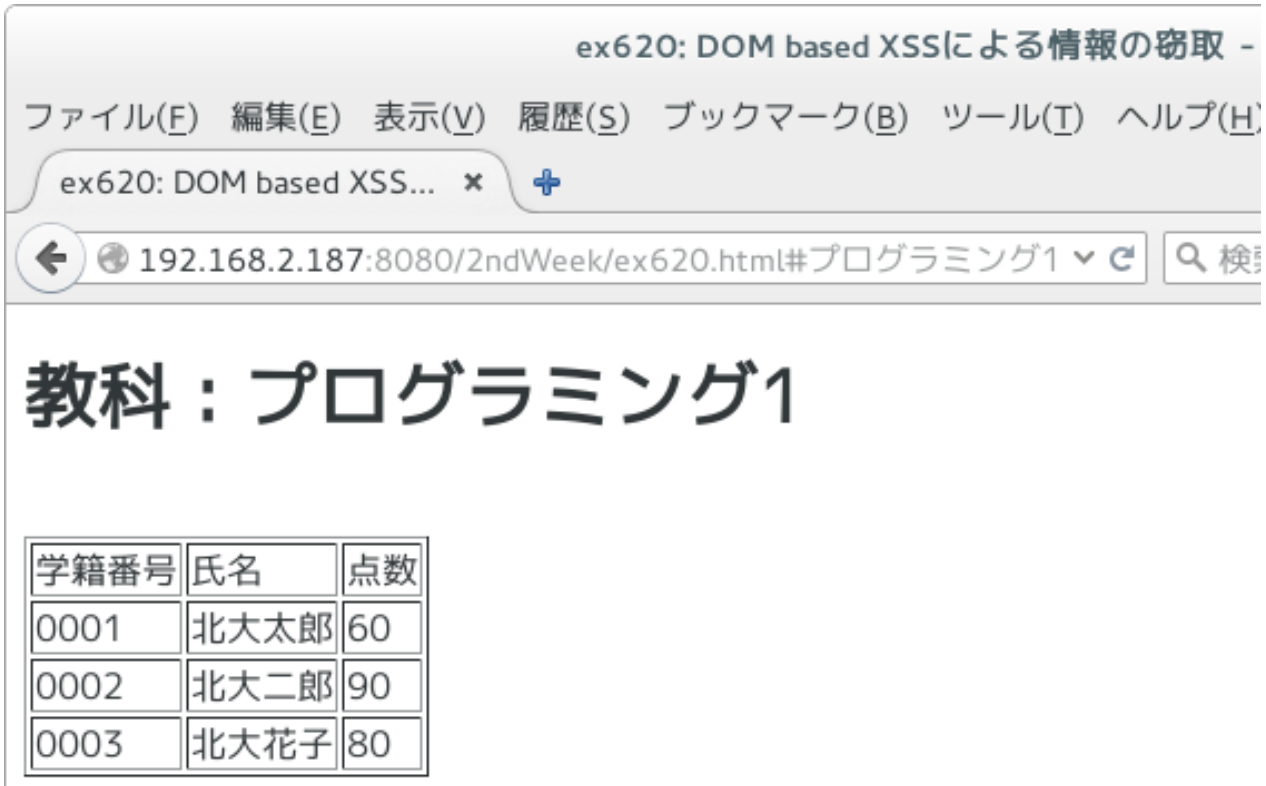
情報理工学実験
モバイルWebアプリケーション

情報基盤センター／飯田勝吉

第６回：JavaScriptのセキュリティ

Web上には重要情報が書かれている



攻撃者が不正なJavaScriptコードをつかえば、重要情報を窃取できるのではないか？

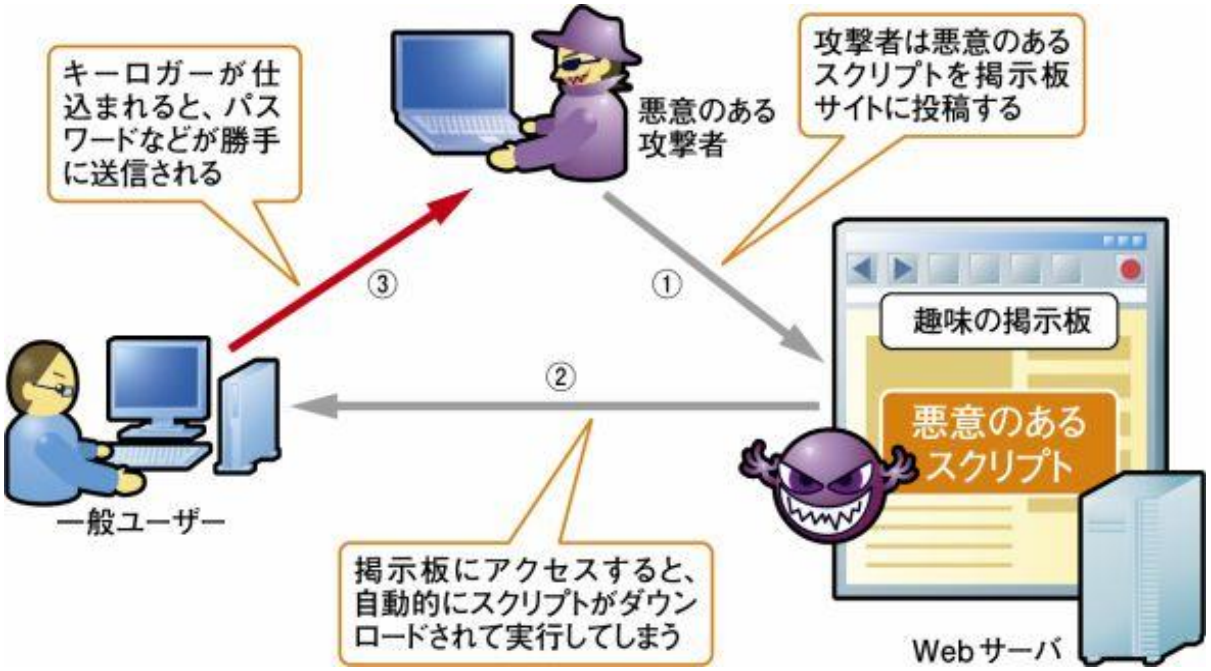
答え

- JavaScriptコードの書き方に注意が足りないと、情報を盗まれるなど、いろいろな危険性がある！

- **危険性を理解し、それを防ぐ方法を知る必要がある！**

クロスサイトスクリプティング攻撃と脆弱性

- Cross Site Scripting (XSS)
  - ユーザ、Webサイトの開発者JavaScriptの開発者が意図していない不正なコードを攻撃対象者のブラウザ上で実行させる攻撃。また、そのような攻撃を生じさせる脆弱性。



http://ascii.jp/elem/000/000/423/423682/index-2.htmlより引用

クロスサイトスクリプティング攻撃の種類

- サーバ側のプログラムに原因がある場合
  - Webサーバで実行されるPHPなどのプログラムが脆弱で結果的に不正なJavaScriptコードをユーザのブラウザ上で実行されるかもしれない
- ブラウザ側のプログラム(JavaScript)に原因がある場合
  - JavaScriptの正当な利用者が、悪意のある第三者に攻撃されるかもしれない
- **安全なJavaScriptコードを学ぶ必要がある**

クロスサイトスクリプティング攻撃による被害

- １．攻撃対象者のブラウザ等に不正コードをインストールさせる
  - キーロガーがインストールさせられた場合
    - パスワードなど重要情報が漏えい
  - ボットプログラムがインストールさせられた場合
    - 他者を攻撃する際に攻撃者の「手下」にされる
- ２．攻撃対象者のブラウザを介して機密情報を窃取する
- ３．攻撃対象者のブラウザで表示されている情報を不正に書き換える
  - オンラインバンキング等のWeb上に表示されている重要情報の不正な買い替えが可能

不正JavaScriptコードを使った機密情報の窃取

- Web画面に表示されている情報の窃取
  - 電子カルテの情報
  - 成績情報
  - 入試出願状況に関する情報
  - など
- Web画面に表示されていない情報の窃取
  - クッキー
    - ログインが成功したことを示す、ブラウザが記憶している乱数列の情報など
    - クッキーが窃取されると、攻撃者に不正ログインを許すかもしれない

機密情報の窃取を防ぐ必要がある

初回ガイダンススライドの再掲 課題の補足訂正（２／３）

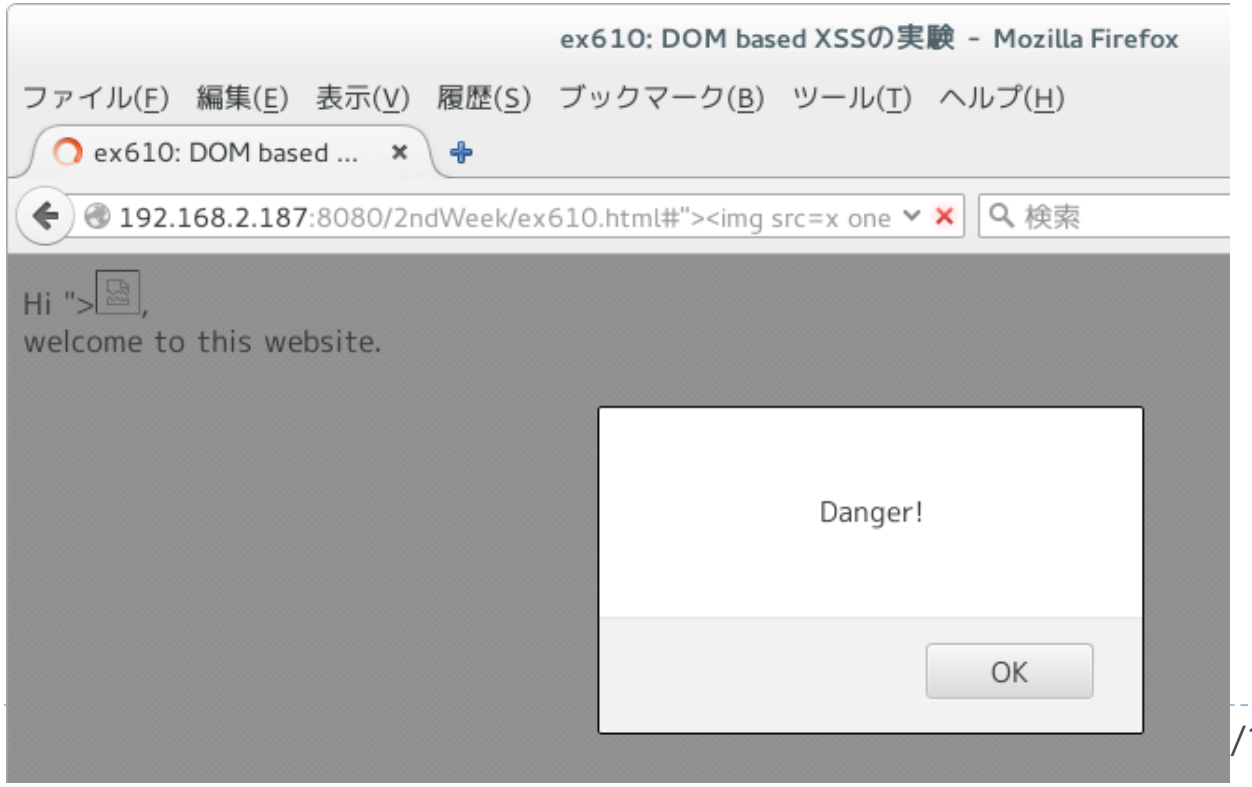
- 第６回課題 (pp.99-102, pp.114-118)
  - Firefoxで演習を行ってきたが、Firefoxのバージョンが上がり、脆弱性がふさがれたため、演習ができなくなった
  - そのため、脆弱性のあるFirefoxを別途用意した
    - 用意した古いFirefox ＝ Firefox38.8.0 (ESR)
    - `/home/work/webapp/firefox/firefox`
  - 以下は詳細説明
  - 上記Firefoxの利用方法
    - `~/.bashrc`に以下の行を追加
    - `alias f-old="/home/work/webapp/firefox/firefox"`
    - 端末アプリケーション上で “`f-old`” をタイプすることで実行可能
  - 注意事項
    - 初回起動時にアドオンに関する問い合わせがでるが、「次へ」「完了」をクリックすればよい
    - 通常のFirefoxと同時起動はできないので、通常のFirefoxを終了してから起動すること

課題６－１(ex610.html)

課題6-1
特殊な記号から始まるURLの末尾部分

- URLパラメータ
  - http://example.co.jp/**?name=Iida&year=2016**
  - **赤字部分**をURLパラメータとよぶ
  - **サーバ**が処理をする
- URLフラグメント
  - http://example.co.jp/**#38943928798**
  - **赤字部分**をURLフラグメントとよぶ
  - **ブラウザ(JavaScript)**が処理をする
- **URLフラグメントに不正なコードが紛れ込むと危険**

- 課題６－１：不正コード
  - http://192.168.x.x:8080/2ndWeek/ex610.html#"><img src=x onerror=alert('Danger!')></img>
    - <img> タグを入れているので、画像を引用しようとする
    - しかし src が存在しないファイルを指しているので、エラーになる
    - エラーになると、onerror= からの命令が実行されてしまう



課題６－１：以下の調査を行え

- (1) 実験室の端末にインストールされている２種類のブラウザFirefox, Chrome上で不正コードを埋め込んでex610.htmlを実行することで、ブラウザ上でJavaScriptが実行されるかどうかを調査せよ。レポートには、調査結果を示す画像を添付すること。
  - Firefoxはf-oldだけでよい。

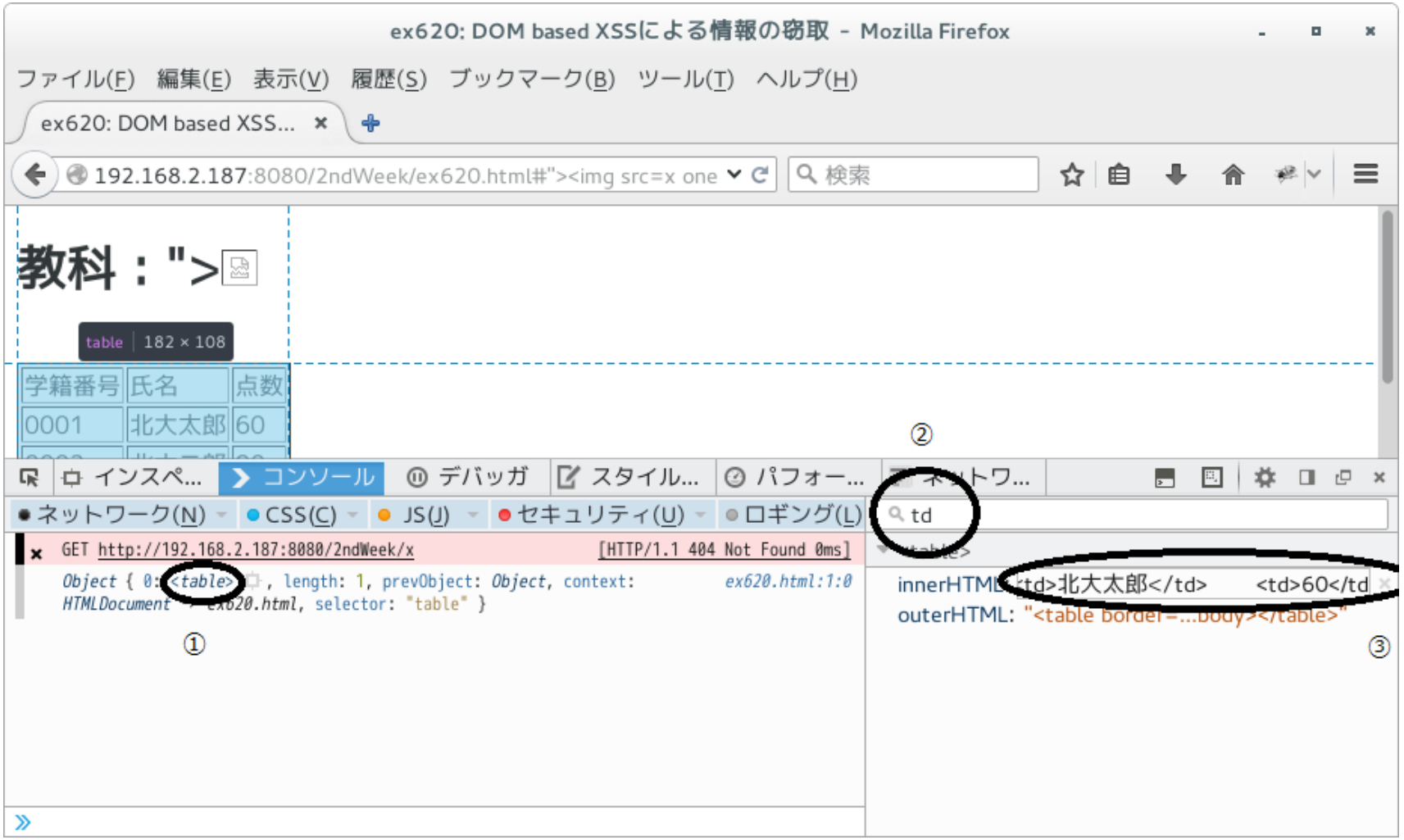
- (2) (発展課題) 私物スマホ、タブレット上で(1)と同様の調査を行うこと。レポートには、調査結果を示す画像を添付すること。

- 私物スマホを持っている人：すべてのブラウザで試す必要はありません
- 持っていない人：もっていないとこの課題はできないので、もっていないとレポートに書いて下さい

課題６－２

- 不正コードをつかえばex620.htmlの成績情報の搾取ができる。
  - ここでは、実際に成績情報を外部に送るのではなく、一般のブラウザ利用者が気が付かないコンソール画面を外部とみなして、以下のコードで攻撃する
  - http://192.168.x.x:8080/2ndWeek/ex620.html#"><img src=x onerror=**console.log(#)**></img>
- (1)console.log(#)の#部分を書き直し、成績情報をコンソール画面に出力するURLを導き出せ。その際、通常のJavaScriptの機能を用いること。
- (2)console.log(#)の#の部分を書き直し、成績情報をコンソール画面に出力するURLを導き出せ。その際、jQueryの機能を用いること。

課題６－２の成功例


ex620: DOM based XSSによる情報の窃取 - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)
ex620: DOM based XSS...
192.168.2.187:8080/2ndWeek/ex620.html#"><img src=x one
検索
教科：">
table | 182 × 108
学籍番号 氏名 点数
0001 北大太郎 60
②
インスペ... コンソール デバッガ スタイル... パフォー... ネットワ...
ネットワーク(N) CSS(C) JS(J) セキュリティ(U) ロギング(L)
td
GET http://192.168.2.187:8080/2ndWeek/x [HTTP/1.1 404 Not Found 0ms]
Object { 0: <table> , length: 1, prevObject: Object, context: HTMLDocument ex620.html, selector: "table" } ex620.html:1:0
innerHTML: <td>北大太郎</td> <td>60</td>
outerHTML: "<table border=...body></table>"
①
③

課題６－２注意

- この実験課題を用意した理由
  - JavaScriptの悪用方法を教えるため　**ではない**
  - 安全なコードを書くためには、どこが脆弱で、どのようにすれば安全になるかを学習する　**ため**

- レポートに記載すべき事項
  - この実験課題で学習したことを悪用せず、不正アクセス禁止法などの法律違反を行なわない

  - ことを示す宣誓の文言

課題６－３

- 課題6-3は演習が継続できないため、やらなくてよい

- 過去のサンプルプログラムex420.htmlは脆弱性を含んだコードである

ソース

例：URLフラグメント、
外部APIからの出力、
外部HTML

シンク

例：JavaScriptの
innerHTMLへの代入

課題６－３
安全なプログラムを書くためのガイド

- １）不正な入力が入り込む可能性のある「ソース」を意識する
- ２）「ソース」からのデータを出力する際は、エスケープ処理・またはサニタイジング処理を行う
- ３）「シンク」は安全な関数を用いる
- ４）ライブラリを用いる場合は、最新版を利用する。

- 詳しくは１１７～１１８ページをみてください

課題６－３

- 過去のサンプルプログラム**ex420.html**は脆弱性を含んだコードである。本資料117～118ページに記載したセキュアプログラミングのガイドを参考に、安全なコードに書き直すこと。

- 注意事項
  - ex410-load.htmlが不正に書き換えられる可能性がある場合は、ex410.htmlも脆弱なコードになりうるので、その場合はex410.htmlも書き直す必要がある

- レポート記載事項
  - 画面添付は不要です。ソースコードのみを添付してください

課題６－３（再掲）

- 過去のサンプルプログラムex420.htmlは脆弱性を含んだコードである

ソース

例：URLフラグメント、
外部APIからの出力、
外部HTML

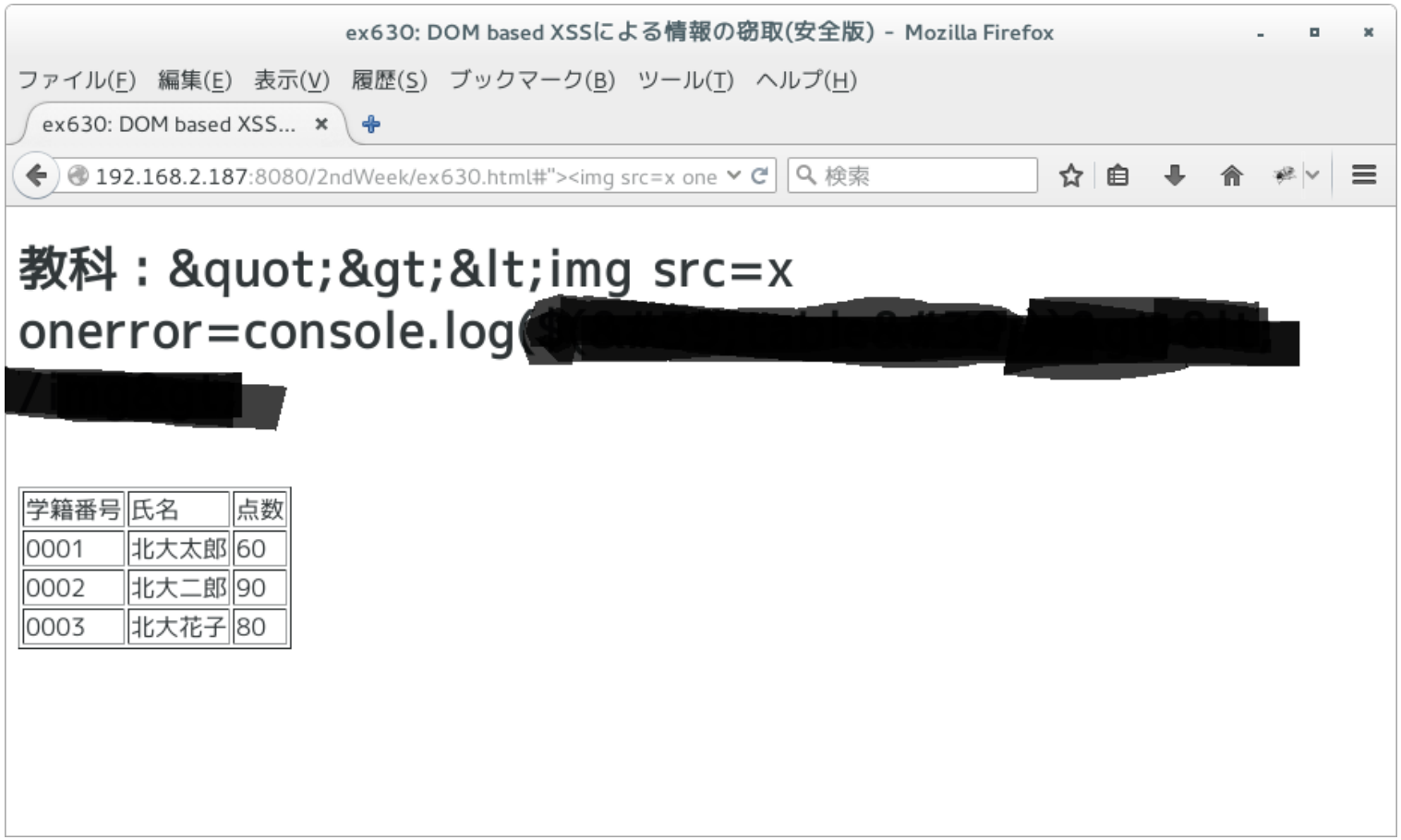
ex420.html
ex410.html

シンク

例：JavaScriptの
innerHTMLへの代入

課題６－３補足

- ex630.html = ex620.htmlの安全版



課題６－４（発展課題）

- DOM based XSSによる被害を軽減するためにはどのような活動を行うことが有効かを考察せよ。具体的には、巻末の解説に記載した安全なコードの原則を自分なりにまとめ、そのうえで、脆弱なコードを減らし、安全なコードを増やしていくために
  - プログラマに対する教育活動
  - 技術的に安全にする方法の研究開発
- 等の活動を行うことが必要となる。これらのことを考察し、レポートに記載すること。