問題１

下線は DOM のもの、それ以外は JavaScript のもの

```
var anchorTags = document.getElementsByTagName("a") ;
for (var i = 0; i < anchorTags.length ; i++)
{
alert("Href of this a element is : " + anchorTags[i].href + "¥n");
}
```

正確な記述

document.getElementsByTagName("a")→

window.document.getElementsByTagName("a")

解説

このコードは JavaScript エンジンによってどんな処理がされ、DOM のパーサに何が送られたのかを教えてくれます。

JS(JavaScript)	DOM	解説
var anchorTags =		これにより "anchorTags" という名の JavaScript (JS) 変数が生成されます。
	document.getElementsByTagName("a")	"document" インターフェイスは DOM1 コアで真っ先に定義されたインターフェイスです。document はページの全てを納めています。DOM1 コアは getElementsByTagName() メソッドを "document" インターフェイスに定義しました。これは "ノードリスト"(ノードを納めた一種の DOM 専用の配列) 中から関数に渡された引数にマッチする全てのタグをドキュメントソースに現れた順序で取得します。これで "anchorTags" 変数はノードリストとなります。
;		基本的な命令終了セミコロン。JS によるもの。
for (var i = 0; i <		プログラミング言語の典型的 "for" ループ。これで anchorTags ノードリスト中に格納された各ノードを処理していけます。このループのスコープ中の変数 "i" 宣言に注目。これも JS。
	anchorTags.length	DOM1 コアは "NodeList" インターフェイスの "length" プロパティを定義しています。これはノードリスト中に格納されるノードの数を示す整数を返します。JS に array::length が

		あることに注目。
; i++)		典型的な JS 整数変数の１インクリメント。
{ alert("Href of this a element is : " +		alert() は渡した引数(文字列)のダイアログをポップアップさせるメソッドです。そして次に "+" 演算子を用いて単純な文字列結合を利用します。
	anchorTags[i].href	"href" は DOM1 HTML 仕様 で定義された HTMLAnchorElement インターフェイスのプロパティです。これはアンカー要素に定義された "href" 属性が存在する場合その属性値を返します。anchorTags[i] も使用しましたが、実のところ配列の i 要素にアクセスする JS 流の方法です。これを行う正しい "DOM 流" はノードリストインターフェイスに定義されている "item()" メソッドを使用することです。つまりこのように: anchorTags.item(i).href 誰もこんな事はしませんが。
+ "¥n");		更に文字列結合。文字列の終わりにキャリッジリターン([訳註] 厳密には、文字コードに依ります) を挿入します。

問題２

レベル 1

DOM レベル 1 仕様はコアと HTML との 2 つの部分に分けられています。コア DOM1 節では、XML ドキュメントを表す拡張インターフェイスの定義だけでなく任意の構造化された文書を表現できる、低レベルな基本的インターフェイス一式を用意しています。HTML レベル 1 節では HTML ドキュメント用により手軽に使えるビューを提供するため、コアレベル 1 で定義される基本的インターフェイスと併用する、追加のより高レベルなインターフェイスを用意しています。 DOM1 で導入されるインターフェイスには、中でも特に、Document, Node, Attr, Element, Text インターフェイスが挙げられます。全てのインターフェイスは XML や HTML ドキュメントとのやり取りに使用する属性 及び/或いは メソッドを持っています。

サポートレベル: 優秀。 コア 及び HTML 仕様関連のバグを参照。

レベル 2

DOM レベル 2 仕様というのは実際には、DOM2 コア、ビュー、イベント、スタイル、

探索と範囲(Traversal and Range)、HTML という 6 つの別々の勧告です。これを書いている時点では DOM2 HTML はまだ W3C のワーキングドラフトです。[訳註] 和訳時点では勧告となっています。 レベル 2 の大半は Mozilla でサポートされています。

DOM2 コア は DOM1 コアの機能を拡張したもので、XML 用に特化したインターフェイスも含んでいます。 DOM2 コアで導入されたメソッドには例えば有名な getElementById や多くの名前空間関連のメソッドがあります。
サポートレベル: 大変宜しい。 コア 仕様関連のバグを参照。

DOM2 ビュー によりプログラムやスクリプトはドキュメントを表す内容を動的にアクセス及び更新できます。導入されたインターフェイスは AbstractView と DocumentView です。
サポートレベル: 完璧。

DOM2 イベント はプログラムやスクリプトに汎用イベントシステムを提供します。イベントフロー、キャプチャ、バブリング、キャンセルの概念を導入しています。有名なメソッドとしては addEventListener や handleEvent があります。幾つかのインターフェイスはイベントの取り扱いを快適なものにしてくれます: EventTarget, EventListener, Event, DocumentEvent, MouseEvent, MutationEvent などです。但しキーボード操作イベントに使用するインターフェイスは含まれていません。これは以降のバージョンの DOM で扱われます。
サポートレベル: 大変宜しい。 イベント 仕様関連のバグを参照。

DOM2 CSS や DOM2 スタイル ([訳註] DOM スタイルシート と DOM CSS の 2 部構成となっている DOM2 スタイル仕様の各部のこと)はプログラムやスクリプトによるスタイルシートドキュメントの内容への動的アクセス及び変更を可能にします。これにはスタイルシート、カスケーディングスタイルシート、CSSRule、CSSStyleDeclaration、有名な getComputedStyle (Mozilla でサポートされています！)、とっても多くの CSS2Properties、そして考え得る限りあらゆるメディアルールのインターフェイスが含まれます。
サポートレベル: 良い。 スタイル 仕様関連のバグを参照。

DOM2 探索と範囲(Traversal and Range)によりプログラムやスクリプトはドキュメント中を動的に移動したりコンテントの範囲を一意指定したりできます。 DOM2 探索では NodeIterator や TreeWalker といった、ドキュメントのコンテント中を簡単に辿るためのインターフェイスを提供します。 DOM2 範囲 ではドキュメント、ドキュメント断片、

或いは属性中の範囲に対してコンテントを生成、挿入、変更、削除できるようにします。これは一組の境界点の間のコンテント全体を選択するものと考えることができます。
サポートレベル: 範囲については基本部分、探索については一切なし。

DOM2 HTML はプログラムやスクリプトが HTML ドキュメントの構造に対する動的アクセス及び変更を可能にします。DOM2 Core の可能性を基に DOM1 HTML で定義されたインターフェイスを拡張するものであり、contentDocument プロパティやフレームに含まれるドキュメントへの手軽なアクセス方法を導入しています。
サポートレベル: 良い。 HTML 仕様関連のバグを参照。

レベル 3
DOM レベル 3 仕様は現時点では 3 つに分けられた勧告からなります。 DOM レベル 3 コア、DOM レベル 3 内容モデル及び読み込みと保存、DOM レベル 3 イベントの 3 つです。これを書いている時点では、3 つ全てまだワーキングドラフトです。([訳註] DOM3 の構成はその後大きく変わっていますので、この節の記述は参考程度と考えて下さい)
DOM レベル 3 コアは DOM1 及び DOM2 コア仕様を拡張します。新しいメソッドやプロパティとしては adoptNode(), strictErrorChecking, textContent といったものが挙げられます。
サポートレベル: baseURI プロパティだけがサポートされています。

DOM3 内容モデル及び読み込みと保存 ではプログラムやスクリプトがドキュメントのコンテント、構造及びスタイルに動的にアクセス及び変更できるようになります。この仕様は DOM3 コア 実装に依存しています。非常に強力なものですが、今のところ私には複雑すぎます。もっと理解したら再度報告します。([訳註] 和訳時点までに、抽象構造(Abstract Schemas)及び読み込みと保存 と名前を変えた後更に抽象構造部を放棄(？)して読み込みと保存だけになっています)
サポートレベル: なし。

DOM3 イベントは DOM2 イベント仕様の拡張です。これは Mozilla DOM チームで作業している Tom Pixley によって為されており、良い感じです！ この仕様では主にキーボードイベントとその取り扱い方に焦点を当てています。遂に！
サポートレベル: なし。