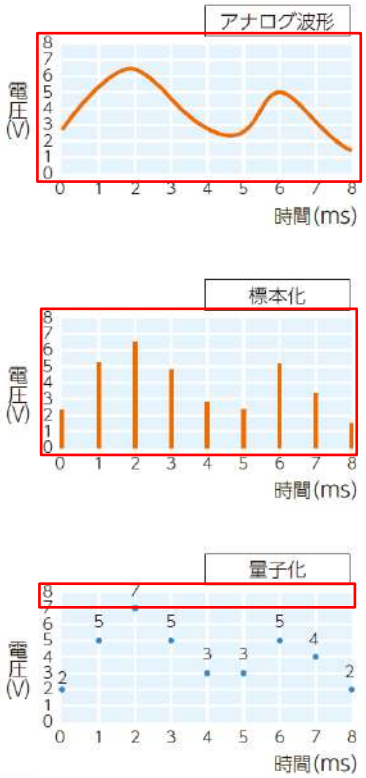
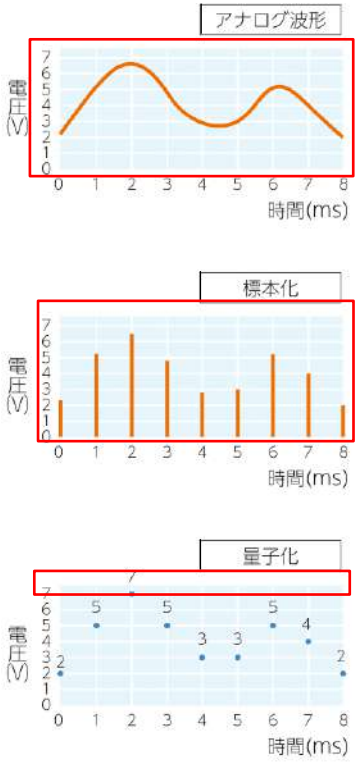
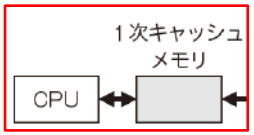
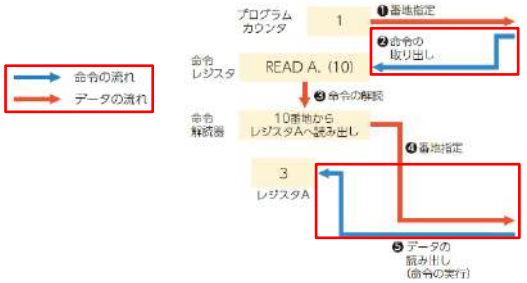
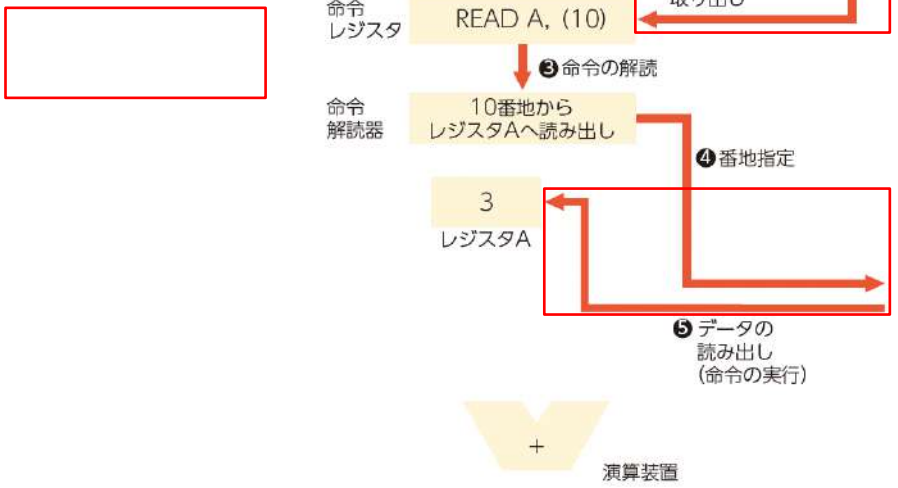
図書の記号･番号　情Ⅰ703

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
1	19	例題1 問(1)	同級生にあげ	多数の同級生 に配布し
2	47	19行	⑦$00000001_{(2)}$ (→補数)　　⑧$01010011_{(2)}$ (→補数)	⑦$00000001_{(2)}$ → 補数　　⑧$01010011_{(2)}$ → 補数
3	51	1-4行	次の計算結果を指定された形式で表現しなさい。ただし数値は4ビットの整数とし，2進数の負の値は補数で表現すること。 ①$10_{(2)}+A_{(16)}$→10進数　②$100_{(2)}-C_{(16)}-5_{(10)}$→2進数 ③$B_{(16)}-11_{(10)}-11_{(2)}$→16進数　④$7_{(10)}+3_{(16)}-1100_{(2)}$→2進数	次の計算をしなさい。ただし数値は4ビットの整数とし，減算は補数を使って計算すること。 ①$0100_{(2)}-0011_{(2)}$　②$1110_{(2)}-0010_{(2)}$ ③$1101_{(2)}-0101_{(2)}$　④$1111_{(2)}-1100_{(2)}$

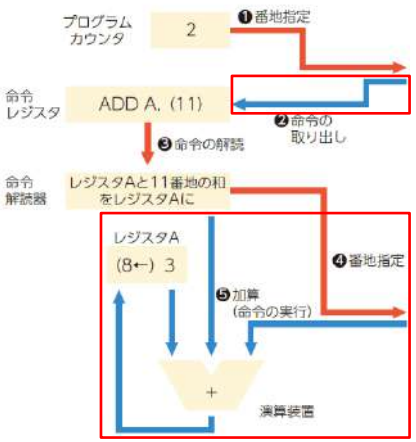
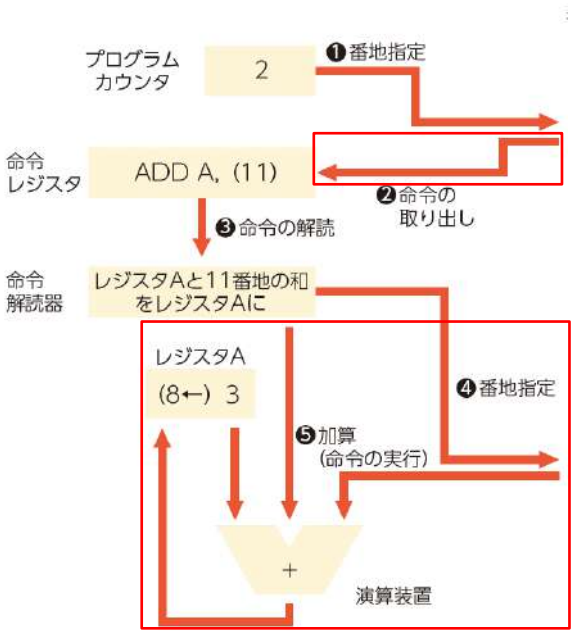
図書の記号・番号　情Ⅰ703

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
4	54	図2		

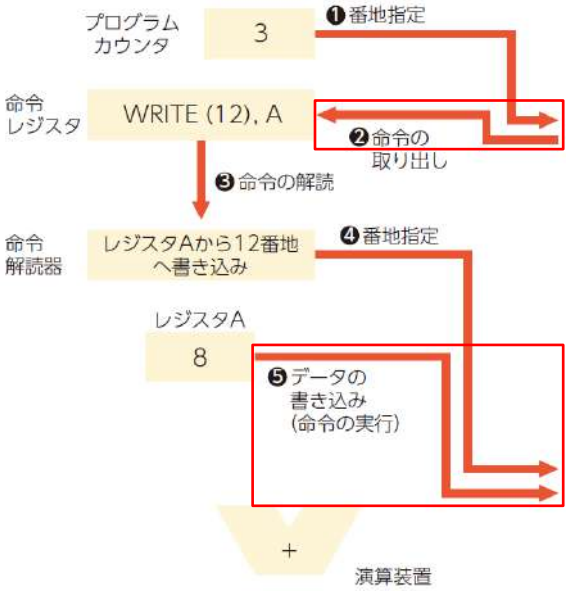
図書の記号・番号　情Ⅰ703

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
5	61	図3		
6	61	図3 キャプション	キャッシュメモリの仕組み	CPU，1次/2次キャッシュメモリと主記憶装置の例
7	62	手順1 図		

図書の記号・番号　情Ⅰ703

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
8	63	手順2 図		

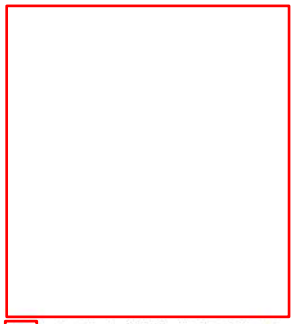
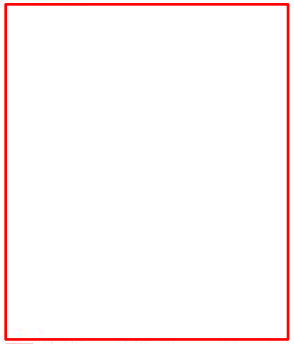
図書の記号・番号　情Ⅰ703

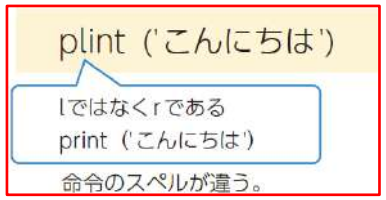
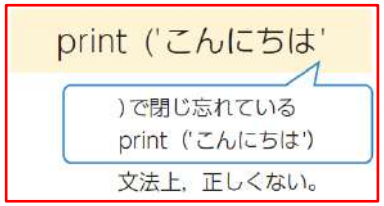
番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
9	63	手順3 図		

図書の記号・番号　情Ｉ703

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
10	68	3-4行	値を量子化してから2ビットに符号化して	値を2ビットで量子化してから符号化し,
11	91	11行	parity bit パリティビット	check digit チェックディジット
12	91	13行	パリティビットを 削除	（削除）
13	91	17行	削除	（削除）
14	91	17行	。ただし	。この1ビットをparity bitパリティビットという。ただし
15	93	6-7行	公開鍵暗号方式で暗号化・復号を行う二重の暗号化方式を利用している。	公開鍵暗号方式または鍵共有方式で共有する方法をとっている。

図書の記号・番号　情Ⅰ703

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
16	93	側注⑤および⑥	 ❺もともと利用されていたSSLは，セキュリティ強化のために改良されTLSに移行したが，SSLという語が普及しているためその発展系という意味でSSL/TLSと表記したり，単にSSLと記される場合もある。  ❻デジタル署名だけでは，な	❺共通鍵暗号方式で利用する共通鍵を安全に共有する方法の一つである。大まかな手順は次の通りである。 1. 送信側，受信側でそれぞれ秘密鍵を準備する。 2. 送信側，受信側でそれぞれ秘密鍵を利用して公開鍵を計算し，相手に送る。 3. 送信側，受信側はそれぞれ受け取った公開鍵と自分がもつ秘密鍵を使用し，共通鍵を算出するための数値を求める。 4. 3で共有された数値を使い，送信側も受信側も等しい値の共通鍵を算出する（鍵共有完了）。 ❻もともと利用されていたSSLは，セキュリティ強化のために改良されTLSに移行したが，SSLという語が普及しているためその発展系という意味でSSL/TLSと表記したり，単にSSLと記したりする場合がある。 ❼デジタル署名だけでは，な
17	93	図4キャプション	SSL／TLS	SSL／TLS（共通鍵を公開鍵暗号方式で共有する場合）

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
18	93	11行	❻	❼
19	93	13行	❼	❽
20	93	側注⑦	❼	❽
21	128	9行	データの処理	データの扱い
22	133	構文エラーの例		
23	134	表1	別添 1	別添1

図書の記号･番号　情Ⅰ703

<table>
<tr><th rowspan="2">番号</th><th colspan="2">訂正箇所</th><th>原　　文</th><th>訂　　正　　文</th></tr>
<tr><th>ページ</th><th>行</th><th></th><th></th></tr>
<tr><td>24</td><td>135</td><td>表2</td><td>
<table>
<tr><th colspan="2">算術・結合演算子</th><th colspan="2">比較演算子</th><th colspan="2">代入演算子</th></tr>
<tr><td rowspan="2">＋</td><td>数値の足し算</td><td>==</td><td>等しい</td><td>=</td><td>代入</td></tr>
<tr><td>文字列の結合</td><td>!=</td><td>等しくない</td><td colspan="2">論理演算子</td></tr>
<tr><td>－</td><td>引き算</td><td>＜</td><td>未満</td><td>and</td><td>かつ</td></tr>
<tr><td>＊</td><td>掛け算</td><td>＞</td><td>超過</td><td>or</td><td>または</td></tr>
<tr><td>／</td><td>割り算</td><td><=</td><td>以下</td><td>not</td><td>ではない</td></tr>
<tr><td>％</td><td>割り算の余り</td><td>>=</td><td>以上</td><td colspan="2">ドット演算子</td></tr>
<tr><td colspan="4"></td><td>.</td><td>～の</td></tr>
</table>
</td><td>
<table>
<tr><th colspan="2">算術・結合演算子</th><th colspan="2">比較演算子</th><th colspan="2">代入演算子</th></tr>
<tr><td rowspan="2">＋</td><td>数値の足し算</td><td>==</td><td>等しい</td><td>=</td><td>代入</td></tr>
<tr><td>文字列の結合</td><td>!=</td><td>等しくない</td><td colspan="2">論理演算子</td></tr>
<tr><td>－</td><td>引き算</td><td>＜</td><td>未満</td><td>and</td><td>かつ</td></tr>
<tr><td>＊</td><td>掛け算</td><td>＞</td><td>超過</td><td>or</td><td>または</td></tr>
<tr><td>／</td><td>割り算</td><td><=</td><td>以下</td><td>not</td><td>ではない</td></tr>
<tr><td>％</td><td>割り算の余り</td><td>>=</td><td>以上</td><td colspan="2">ドット演算子</td></tr>
<tr><td colspan="4"></td><td>.</td><td>～の</td></tr>
</table>
</td></tr>
<tr><td>25</td><td>171</td><td>8行</td><td>込み関数などを利用する。これらの関数</td><td>込みメソッドなどを利用する。これらのメソッド</td></tr>
<tr><td>26</td><td>171</td><td>図2
右表

図3
右表

図4
右表

図5
右表</td><td>関数

関数

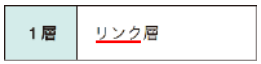
関数

関数</td><td>メソッド

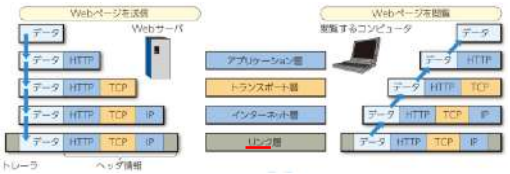
メソッド

メソッド

メソッド</td></tr>
</table>

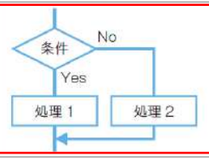
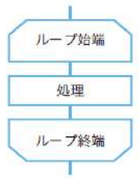
番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
27	171	図2キャプション	関数の例	メソッドの例
		図3キャプション	関数の例	メソッドの例
		図4キャプション	関数の例	メソッドの例
		図5キャプション	関数の例	メソッドの例
28	172	13行	関数	メソッド
29	179	3行	60kg重	588N
30	179	4行	kg重	N
		6行	kg重	N
31	72	表1		ネットワークインタフェース

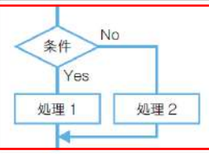
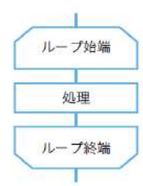
図書の記号･番号　情Ⅰ703

番号	訂正箇所 ページ	訂正箇所 行	原　　文	訂　　正　　文
32	73	上の図		ネットワークインタフェース
33	73	側注④	❹アプリケーション層のヘッダは，全パケットには付加されない。また，リンク層では，ヘッダだけでなくデータの最後にエラーをチェックするためのトレーラという情報も付く。	ネットワーク インタフェース
34	190	左から4段目29-30行	ネットワーク……70 ネットワーク層……72	ネットワークインタフェース層・・・・・72
35	191	左から3段目22行	リンク……37 リンク層……72 類似色……26　削除	リンク……37 類似色……26

訂正申請別添用紙

図書の記号・番号	情 I 703

<table>
<tr><th>別添 No.</th><th>原　　文</th><th>訂　　正　　文</th></tr>
<tr><td>1</td><td>
<table>
<tr><td>順次</td><td> 
</td><td>処理1
処理2</td><td>上から下へ記述された順に（ここでは処理１の後に処理２）処理を実行する。</td></tr>
<tr><td>選択</td><td> 
</td><td>if 条件:
　処理1
else:
　処理2</td><td>条件が真(Yes)の時（条件が成立する時）は処理１を実行し，偽(No)の時（条件が成立しない時）は処理２を実行する。</td></tr>
<tr><td>反復</td><td> 

※Pythonにはループ終端に相当する文（命令，コード）がない。ループの範囲は字下げで示される。</td><td>
<table>
<tr><td>①</td><td>for 変数 in range(回数):
　処理</td><td rowspan="3">④</td><td rowspan="3">初期設定
while 条件:
　処理
　再設定</td></tr>
<tr><td>②</td><td>for 変数 in range(値1, 値2, 増減値):
　処理</td></tr>
<tr><td>③</td><td>for 変数 in 配列:
　処理</td></tr>
</table>
処理を繰り返す。途中で繰り返し（ループ）を終了するにはbreakを入れる。
①変数の値を０から［回数－１］の範囲まで１つずつ変化させながら処理を繰り返す。範囲外になるとループを抜ける。
②変数の値を値１から値２までの範囲（値２は含まれない）で，増減値の幅で変化させながら処理を繰り返す。範囲外になるとループを抜ける。
③変数に配列（リスト）の要素を一つずつ取り出しながら処理を繰り返す。要素をすべて取り出すとループから抜ける。
④条件が真の間，ループの始端と終端の間にある処理を繰り返し，偽になるとループから抜け出す。初期設定では，反復処理の中で使用するループ変数に初期値を設定する。再設定では，ループ変数の値を増減する。
</td></tr>
</table>
</td><td>
<table>
<tr><td>順次</td><td> 
</td><td>処理1
処理2</td><td>上から下へ記述された順に（ここでは処理１の後に処理２）処理を実行する。</td></tr>
<tr><td>選択</td><td> 
</td><td>if 条件:
　処理1
else:
　処理2</td><td>条件が真(Yes)の時（条件が成立する時）は処理１を実行し，偽(No)の時（条件が成立しない時）は処理２を実行する。</td></tr>
<tr><td>反復</td><td> 

※Pythonにはループ終端に相当する文（命令，コード）がない。ループの範囲は字下げで示される。</td><td>
<table>
<tr><td>①</td><td>for 変数 in range(回数):
　処理</td><td rowspan="3">④</td><td rowspan="3">初期設定
while 条件:
　処理
　再設定</td></tr>
<tr><td>②</td><td>for 変数 in range(値1, 値2, 増減値):
　処理</td></tr>
<tr><td>③</td><td>for 変数 in 配列:
　処理</td></tr>
</table>
処理を繰り返す。途中で繰り返し（ループ）を終了するにはbreakを入れる。
①変数の値を０から［回数－１］の範囲まで１つずつ変化させながら処理を繰り返す。範囲外になるとループを抜ける。
②変数の値を値１から値２までの範囲（値２は含まれない）で，増減値の幅で変化させながら処理を繰り返す。範囲外になるとループを抜ける。
③変数に配列（リスト）の要素を一つずつ取り出しながら処理を繰り返す。要素をすべて取り出すとループから抜ける。
④条件が真の間，ループの始端と終端の間にある処理を繰り返し，偽になるとループから抜け出す。初期設定では，反復処理の中で使用するループ変数に初期値を設定する。再設定では，ループ変数の値を増減する。
</td></tr>
</table>
</td></tr>
</table>