



# C Piscine

## Shell 01

*Summary:* このドキュメントは、42の *C Piscine Shell 01* モジュール用の課題です。

*Version:* 8.1

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>AI Instructions</b>	<b>3</b>
<b>III</b>	<b>Foreword</b>	<b>6</b>
<b>IV</b>	<b>Exercise 00 : Exam</b>	<b>7</b>
<b>V</b>	<b>Exercise 01 : print_groups</b>	<b>8</b>
<b>VI</b>	<b>Exercise 02 : find_sh</b>	<b>9</b>
<b>VII</b>	<b>Exercise 03 : count_files</b>	<b>10</b>
<b>VIII</b>	<b>Exercise 04 : MAC</b>	<b>11</b>
<b>IX</b>	<b>Exercise 05 : Can you create it ?</b>	<b>12</b>
<b>X</b>	<b>Exercise 06 : Skip</b>	<b>14</b>
<b>XI</b>	<b>Exercise 07 : r_dwssap</b>	<b>15</b>
<b>XII</b>	<b>Exercise 08 : add_chelou</b>	<b>16</b>
<b>XIII</b>	<b>Submission and peer-evaluation</b>	<b>17</b>

# Chapter I

## Instructions

- 課題は難易度が低いものから高いものへと難易度順に並んでいます。前の課題が正しく解けていない場合、後の課題が解けていても評価されません。
- ファイルとディレクトリへの権限があることを事前に確認してください。
- すべての課題で、提出手順に従ってください。
- 提出した課題の評価（レビュー）は、あなたの周りにはいるPiscine受験者によって行われます。
- さらに、**Moulinette**という自動評価プログラムでもチェック・採点されます。
- **Moulinette**は非常に厳格かつ厳密に評価を行います。完全に自動化されており、交渉の余地はありません。思わぬ減点を避けるためにも、細部まで丁寧に取り組んでください。
- すべてのシェルの課題は、必ず/bin/shで実行できるようにする必要があります。
- 課題で指定されたファイル以外は、ディレクトリに絶対に残さないでください。
- 質問がある場合は、まず右隣の仲間に聞いてみましょう。いなければ左隣の仲間に聞いてみてください。
- あなたの助けになる参考資料はGoogle / man / インターネット / ...です。
- 課題の例（出力例）をよく読んでください。課題の説明文だけではすぐに分からない要件が隠れていることがあります。

# Chapter II

## AI Instructions

### ● Context

C Piscineは強烈な体験です。42で迎える最初の大きな挑戦として、問題解決、自律性、そしてコミュニティの世界に深く飛び込むことになります。

このフェーズの主な目的は、試行錯誤や反復、そして特にピアラーニングを通じた交流によって、自分の土台を築き上げることです。

AI時代において、近道を見つけることは簡単です。しかし、AIの利用が本当にあなたの成長を助けているのか、それとも真のスキルを身につける上で障害となっていないか、よく考えることが大切です。

Piscineはまたとない人との繋がりを経験する場でもあります。今のところ、それに取って代わるものは何ともありません。AIでさえも。

学習ツールとして、ICTカリキュラムの一環として、そして労働市場での期待の高まりに答えるものとして、私たちのAIに対する考え方の詳細については、Intranet上の専用FAQをご覧ください。

### ● 主なメッセージ

- 近道をせず、強固な基礎を築く。
- 技術力と実践力を真に養う。
- 真のピアラーニングを体験し、学び方や新たな問題の解決方法を学び始める。
- 結果よりも学習の過程が重要である。
- AIに伴うリスクを学び、一般的な落とし穴を避けるための効果的な管理方法と対策を身につける。

## ● 学習者のルール:

- 課題には、AIに頼る前にまず自分の頭で考えること。
- AIに直接的な答えを求めないこと。
- AIに対する42のグローバルな方針を学ぶこと。

## ● このフェーズでの到達目標:

この基礎フェーズを終えることで、以下の点を達成できます。

- 技術とコーディングの適切な基礎を身につける。
- この段階においてAIが危険となりうる理由と、その危険性を理解する。

## ● コメントと具体例:

- 私たちはAIの存在を知っていますし、それがあなたのプロジェクトを解決できることも知っています。しかし、あなたは学ぶためにここにいるのであって、AIが学習したことを証明するためではありません。AIが特定の問題を解決できることを示すためだけに、あなた（や私たち）の時間を無駄にしないでください。
- 42での学習は、答えを知ることではありません。答えを見つけ出す能力を養うことです。AIは直接答えを与えてくれますが、それはあなた自身の論理的思考を妨げます。そして、論理的思考には時間と努力、そして失敗が伴います。成功への道は、決して楽なものではありません。
- Exam中はAIが利用できないことを心に留めておいてください。インターネットもスマートフォンもありません。学習過程でAIに過度に頼りすぎている場合、すぐにそのことに気づくでしょう。
- ピアラーニングは、多様な考え方やアプローチに触れる機会を与え、対人スキルや多角的な思考能力を向上させます。これは、単にボットとチャットするよりもはるかに価値があります。ですから、恥ずかしがらずに、話しかけ、質問し、共に学びましょう！
- AIは学習ツールとして、またそれ自体がトピックとして、カリキュラムの一部になります。独自のAIソフトウェアを開発する機会さえあります。私たちの段階的なアプローチについてさらに学ぶためには、Intranetで利用可能なドキュメントを参照してください。

### ✓ 良い実践例:

新しいコンセプトの学習でつまづいています。近くの人にどうアプローチしたか尋ねてみます。10分ほど話していると、突然ひらめき、理解できました。

**✕ 悪い実践例:**

こっそりAIを使い、良さそうなコードをコピーします。ピアレビューの際には何も説明できず、不合格になりました。ExamではAIがなく、また行き詰まり、不合格になりました。

# Chapter III

## Foreword

Here's what *Wikipedia* says about otters:

The European otter (*Lutra lutra*), also known as the Eurasian otter, Eurasian river otter, common otter and Old World otter, is a European and Asian member of the Lutrinae or otter subfamily, and is typical of freshwater otters.

The European otter is a typical species of the otter subfamily. Brown above and cream below, these long, slender creatures are well-equipped for their aquatic habits. Its bones show osteosclerosis, increasing their density to reduce buoyancy.

This otter differs from the North American river otter by its shorter neck, broader visage, the greater space between the ears and its longer tail.

However, the European otter is the only otter in its range, so it cannot be confused for any other animal. Normally, this species is 57 to 95 cm (23-37 in) long, not counting a tail of 35-45 cm (14-18 in).

The female is shorter than the male.

The otter's average body weight is 7 to 12 kg (15.4-26.4 lbs), although occasionally a large old male may reach up to 17 kg (37 lbs).

The record-sized specimen, reported by a reliable source but not verified, weighed over 24 kg (53 lbs).

The European otter is the most widely distributed otter species, its range including parts of Asia and Africa, as well as being spread across Europe. Though currently believed to be extinct in Liechtenstein and Switzerland, they are now very common in Latvia, along the coast of Norway, and across Great Britain, especially Shetland, which holds 12% of the UK's breeding population. Ireland has the highest density of Eurasian otters in Europe.


In Italy, they can be found in southern parts of the peninsula.

The South Korean population is endangered.

Otters are cute.

# Chapter IV

## Exercise 00 : Exam


	Exercise : 00
	Exam

- 毎週、Intranetで金曜日のExamに登録することを忘れないでください。
- **Exam00**プロジェクトにも必ず登録してください。
- Exam（イベントとプロジェクトの両方）に登録できているかを確認してください。
- Exam（イベントとプロジェクトの両方）に登録できているかを再確認してください。



# Chapter V

## Exercise 01 : print\_groups

	Exercise 01
	print_groups.sh
	提出するディレクトリ : <i>ex01/</i>
	提出するファイル : <b>print_groups.sh</b>
	使用可能な関数 : None

- 環境変数FT\_USERで定義されたユーザーが所属しているグループの一覧を標準出力に出力するコマンドラインを作成してください。
- その際、スペースは入れずカンマで区切ってください。
- 例 :

- FT\_USER=bocal の場合 :

```
$>./print_groups.sh  
bocal,adm,cdrom,sudo,dip,plugdev,lxd,lpadmin,libvirt$>
```

- FT\_USER=daemon の場合 :


```
$>./print_groups.sh  
daemon,bin$>
```



- man id
- 他のPiscine受験生からヒントを得ても、自分の手を動かして作成しましょう！

# Chapter VI

## Exercise 02 : find\_sh

	Exercise 02
	find_sh.sh
	提出するディレクトリ : <i>ex02/</i>
	提出するファイル : <i>find_sh.sh</i>
	使用可能な関数 : None

- サブディレクトリを含む現在のディレクトリ内で、名前が.shで終わるすべてのファイルを検索するコマンドラインを作成してください。
- これらのファイル名を標準出力に出力する際は、.shが付記されていない状態にしてください。
- 出力例 :


```
$>./find_sh.sh | cat -e
find_sh$
file1$
file2$
file3$
$>
```



どんな情報でも鵜呑みにせず、必ず自分でテスト・検証・確認をしましょう！

# Chapter VII

## Exercise 03 : count\_files

	Exercise 03
	count_files.sh
	提出するディレクトリ : <i>ex03/</i>
	提出するファイル : <i>count_files.sh</i>
	使用可能な関数 : None

- サブディレクトリを含む現在のディレクトリ内にある、通常のファイルとディレクトリの合計数を標準出力に出力するコマンドラインを作成してください。
- その際、“.”で始まるディレクトリも含めてください。
- 出力例 :


```
$>./count_files.sh | cat -e
42$
$>
```



失敗も学びの一部です。何度もテストして改善しましょう！

# Chapter VIII

## Exercise 04 : MAC

	Exercise 04
	MAC.sh
	提出するディレクトリ : <i>ex04/</i>
	提出するファイル : <i>MAC.sh</i>
	使用可能な関数 : <i>None</i>


- あなたのコンピュータのMACアドレスを標準出力に出力するコマンドラインを作成してください。その際、アドレスごとに改行してください。



- `man ifconfig`
- 協力し合うことが成功への鍵です！

## Chapter IX

### Exercise 05 : Can you create it ?

	Exercise 05
Can you create it ?	
提出するディレクトリ : <i>ex05/</i>	
提出するファイル : <code>"\?\$*'MaRViN'*\$?\\"</code>	
使用可能な関数 : None	

- “42” のみを含むファイルを作成してください。他の内容は一切含めないでください。
- そのファイルの名前は、以下ようになります。

```
"\?$*'MaRViN'*$?\\"
```

- 出力例 :

```
$>ls -lRa *MaRV* | cat -e
-rw---xr-- 1 75355 32015 2 Oct 2 12:21 "\?$*'MaRViN'*$?\\"$
$>
```

## マイルストーン達成、引き続き頑張りましょう！

この課題の必須問題をすべて完了しました。ここから先、引き続きこの課題の任意問題に取り組むか、次の課題に進むかを選ぶことができます。


- 任意問題に進んでさらに学ぶ。
- 次の課題に進む。

どちらの道に進んでも、役立つ概念を学べます。下記の点を踏まえて判断してください。

- 最初の試験（Exam）や週末に行われるチーム課題（Rush）はC言語が中心です。そのため、事前に学んで経験を積んでおくといいでしょう。
- Piscineの成績は、1つの要素だけではなく様々な観点から評価されます。
  - 各課題の完了状況
  - Piscineの課題全体の進捗度最善の結果につながるように、賢く選択しましょう。
- Piscineが終了するまでの間、同じ課題は、数日から数週間後に再び取り組むことができます。
- 他のPiscine受験生とある程度同じ進捗を保つことで、より円滑に学び合うことができます。

# Chapter X

## Exercise 06 : Skip

	Exercise 06
skip.sh	
提出するディレクトリ : <i>ex06/</i>	
提出するファイル : <i>skip.sh</i>	
使用可能な関数 : None	

- `ls -l` コマンドの1行目から1行おきに、標準出力に出力するコマンドラインを作成してください。
- 出力例 :

```
$>ls -l | cat -e
total 4$
-rw-rw-r-- 1 eagle eagle ** ** 15 15:46 skip.sh$
-rw-rw-r-- 1 eagle eagle ** ** 15 15:41 tata$
-rw-rw-r-- 1 eagle eagle ** ** 15 15:41 titi$
-rw-rw-r-- 1 eagle eagle ** ** 15 15:41 toto$
-rw-rw-r-- 1 eagle eagle ** ** 15 15:41 tutu$
$>
```


```
$>./skip.sh | cat -e
total 4$
-rw-rw-r-- 1 eagle eagle ** ** 15 15:41 tata$
-rw-rw-r-- 1 eagle eagle ** ** 15 15:41 toto$
$>
```



こまめにGitをプッシュしましょう！

# Chapter XI

## Exercise 07 : r\_dwssap

	Exercise 07
	r_dwssap.sh
	提出するディレクトリ : ex07/
	提出するファイル : r_dwssap.sh
	使用可能な関数 : None

- cat /etc/passwd コマンドの出力を、以下の順序で処理するコマンドラインを作成してください。
  - コメントを削除する。
  - 2行目から1行おきに残す。
  - 各ログイン名の文字列を逆順にする。
  - 結果をアルファベット順の降順に並べ替える。
  - 環境変数 FT\_LINE1 と FT\_LINE2 の間のログインのみを残す。
  - それらを1行にまとめ、“,” (引用符を除く) で区切る。
  - 出力の最後は “.” (引用符を除く) で終わる。
- 出力例 (7行目から15行目の場合) :

```
$> ./r_dwssap.sh
sstq_, sorebrek_brk_, soibten_, sergtsop_, scodved_, rlxcm_, rgmecived_, revreswodniw_,
revressta_.$>
```




- 必ず、指示通りの順序で処理されるようにしてください。
- 隣のPiscine受験生に確認してみましたか？



# Chapter XII

## Exercise 08 : add\_chelou

	Exercise 08
	add_chelou.sh
	提出するディレクトリ : <i>ex08/</i>
	提出するファイル : <i>add_chelou.sh</i>
	使用可能な関数 : <i>None</i>

- 以下の処理を行うコマンドラインを作成してください。
  - 変数FT\_NBR1（'\\"?!ベース）と、FT\_NBR2（mrdocベース）から数値を取得する。
  - 両方の合計を、gtaio luSnemfベースで標準出力に出力する。

- 例1 :

```
FT_NBR1=\"?!\"\\  
FT_NBR2=rcrdmddd
```

- 結果 :

```
Salut
```

- 例2 :

```
FT_NBR1=\"\\\"!\\\"\\\"!\\\"\\\"!\\\"\\\"!\\\"\\\"!\\\"\\\"  
FT_NBR2=dcrcmcmoododmrrrmorcmcrmomo
```

- 結果 :

```
Segmentation fault
```



困ったときは、周りにいる誰かに気軽に質問してみましょう！

# Chapter XIII

## Submission and peer-evaluation

課題は、いつも通りGitリポジトリに提出してください。リポジトリ内の提出物のみが、レビュー中の評価対象となります。ファイル名が正しいかどうか、必ず確認してください。



この課題の指示で、明示的に求められたファイルのみを提出してください。