WORD

2010.01 From College of Information Science

```
       -, .-
(   ・ω・ )
○=  {=}○,
|:::::::::\, ',
、、、、し 、、、((  (.@)
```

wwwwwwwwwwwwwwwwwwwwwwwwwwww

13

WORDは仕分けされませんでした号

wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww

CONTENTS

情報学類・情報学群同窓会 心青会のご紹介

文・写真　心青会幹事会　構成　編集部 OB　長田敏之

中学校や高校に同窓会があるように、情報学類・情報学群にも同窓会がある、ということをご存じでしょうか。知らない方はこれを機会に知っていただき、知っている方は、さらに詳しく知っていただきましょう。

◆心青会って何？

筑波大学の情報学類および情報学群の卒業生と教官が集う同窓会です。誕生から十数年、現在 29 期 2,700 名を越える大所帯になっています。会長、副会長、会計監査、事務局、各期幹事で運営しています。他大学等に移られた先生方も、心青会の会員になっていただいております。

▲2006 年大同窓会の集合写真。
以下、2006 年の大同窓会の様子です

◆どんなことをやっている？

隔年で大同窓会を開いて旧交を温め、同窓会員の相互交流を行っています。

また、旧友や恩師への連絡の橋渡しとなるよう、同窓会名簿を発行しています。

さらに、情報学類・情報学群の学生が主体の行事(学園祭出展、謝恩会)への支援も行っています。

▲宴の前に行う心青会総会にて、会長の挨拶

◆これまでの歩み

同窓会の発足は、今から 20 年ほど間にさかのぼります。情報学類設立から 10 年経ち、卒業生や卒業を迎える 10 期生たちの中から同窓会を結成しようという気運が高まりました。

1990 年に第 1 回大同窓会を開催し 1992 年 11 月 28 日の第 2 回大同窓会において、正式

▲テーブルを囲んで同期の輪

な同窓会組織として、心青会が誕生しました。

以来、ほぼ隔年(西暦偶数年)ごとに大同窓会を開催しています。開催地は主に東京都内で行っていますが、前回の 2008 年 10 月 12 日に記念となる第 10 回大同窓会は、つくばで開催しました。大同窓会開催にあたっては、山口情報科学類長や加藤教授をはじめ、多数の先生方にご尽力いただき、100 名を超える OB/OG、先生方が参加されました。

▲お子様連れの方向けに、
キッズコーナーを設けています。

◆今年卒業を迎える学生へ

心青会は、今年卒業を迎える学生のみなさんを新会員として心より歓迎いたします。

年会費などは不要です。情報学類・情報学群同窓生であることを忘れないようにしていただくだけで、十分です。そして、少なくとも同期生間では、お互いに連絡が取れるようにしておきたいと考えております。

学位授与式の日に、幹事会のメンバーが改めてご挨拶に向かいます。

▲ 2006 年はクイズをやりました。
筑波大学や情報産業時事をネタに出題

◆もっと詳しく

心青会についてさらに詳しく知りたい……そんなあなたに、心青会の Web サイトをご用意しています。興味のある方(そうでない人も！)は、是非ご覧ください。過去の同窓会の様子を紹介しているページがありますよ。昔の学位授与式やバグ祭のレポートなんてものもありますよ。

アドレス等は下記をご参照下さい。

▲クイズの景品は、勤め先のグッズを、
みなさん持ち寄ってもらっています。
ボールペン、T シャツ、名刺ケース、
中には海賊王(予定)のぬいぐるみまで！

◆以下、余談の事ながら

なぜ『心青会』なのか。これは、情報の「情」の字を「こざとへん」と「あお」に分けて、心に青の会、となりました。『しんせいかい』と呼ぶのが一般的ですが、その成り立ちからあえて『なさけのかい』と呼ぶ人もいます。

心青会 Web サイト　http://www.jouhou-ob.org/
ご質問・ご意見は　webmaster@jouhou-ob.org　まで

ポケットモンスター廃人日記～厳選編～

文 編集部 IX

■はじめに

2009 年 9 月 12 日、『ポケットモンスター*1　ハートゴールド・ソウルシルバー』が発売されました。初代*2 から通してプレイしている私は歓喜乱舞し、ハートゴールドを入手し、現在プレイ時間は 230 時間を超えました。

なぜ、こんなことになってるかって？

一つ目は、私が一つのゲームをやりこむ人間であるということがあげられます。
二つ目は、このゲームの中毒性にあります。

実は余り知られていませんが、最近のポケットモンスターは低年齢層を対象としたゲーム……の皮を被った**廃人向け**ゲームなのです。

ゆめと　ぼうけんと！
ポケモン　はいじんのせかいへ！
レッツ　ゴー！

■ポケモンがなぜ廃人ゲーなのか

ポケモンが廃人ゲーと呼ばれる所以は、**タマゴ**と**個体値**の存在です。タマゴ、個体値は共に第二世代*3 のころから存在していたシステムですが、仕様がいろいろ変わりましたし、やったことのない人には何のことやらわからないでしょうから解説していきましょう。

---

*1　ポケットモンスター：縮めてポケモン。新バージョンが出るたびにその数は増え続け、現在全 493 種。

*2　初代：赤、緑、青、ピカチュウバージョンのこと。

*3　第二世代：金、銀、クリスタルバージョンのこと。ハートゴールド・ソウルシルバーはこれらをリメイクしたもの。

■タマゴができるまで

まずポケモンのタマゴができるまでについて解説します。

基本的には以下の三つの組み合わせのポケモンを、育て屋[*4]に預けることによってタマゴを入手することができます。

ここでは例としてピカチュウ[*5]のタマゴが出来る組み合わせを示します。

・組み合わせ 1

ピカチュウ♂×ピカチュウ♀　→　ピカチュウ♂ or ピカチュウ♀

一番わかりやすいのがこの組み合わせ。ほしいポケモンと同じ種類の♂と♀を育て屋に預けることによってタマゴを入手することが可能になります。

・組み合わせ 2

マリル[*6]♂×ピカチュウ♀　→　ピカチュウ♂ or ピカチュウ♀

一番お世話になるのがこの組み合わせ。タマゴの中身は♀のポケモンに依存するので、ほしいポケモンと同じ種類の♀と♀とおなじタマゴグループ[*7]の♂を育て屋に預けることによってタマゴを入手することが可能になります。

・組み合わせ 3

メタモン[*8]×ピカチュウ♂ or ピカチュウ♀　→　ピカチュウ♂ or ピカチュウ♀

一番便利なのがこの組み合わせ。メタモンは一部のポケモン[*9]を除いて相手が♂であろうが♀であろうが性別不明であろうがタマゴを入手することが可能なジョーカー的な存在なのです。この組み合わせでないと、♂しか存在しないポケモンや性別不明のポケモンのタマゴを入手することはできません。

---

*4　育て屋：主人公が一歩歩くたびに預けたポケモンの経験値が 1 増える施設。基本料は 100 円で 1 レベル上がるごとに追加料金 100 円が基本料に上乗せされる。値段設定は割と良心的。

*5　ピカチュウ：世界一有名な電気ネズミ。

*6　マリル：みずねずみポケモン。幼馴染キャラが「マリルの　においを　かいだら　**ぞうきん**　みたいな　においが　したの　ちょっと　ショックよねー」 といった内容の電話をかけてくるため、ぞうきんと呼ばれることもある。

*7　タマゴグループ：陸上、怪獣、虫、飛行、妖精、植物、人型、鉱物、不定形、ドラゴン、水中 1、水中 2、水中 3、性別不明、メタモン、タマゴ未発見といった感じにグループ分けされている。これが一致しないとタマゴは出来ない。

*8　メタモン：別名『うむきかい』。

*9　一部のポケモン：タマゴグループ『タマゴ未発見』のポケモン。おもに伝説のポケモン（ホウオウ、ルギアなど）が多い。

■種族値と個体値という概念

ポケモンには隠しパラメータが存在し、ポケモンの種族ごとに違う値を**種族値**、同じ種族でも個体によって違う値を**個体値**と呼びます。

種族値は各種族ごとに決まっていて、素早いマルマイン[*10] のすばやさの種族値が 140 なのに対し、動きの鈍いカビゴン[*11] のすばやさの種族値は 30 といったように定められています。

一方、個体値は個体ごと決められている数値のことで、各ステータス[*12] ごとに 0 ～ 31 の値がランダムに振り分けられています。個体値が異なると同じレベルの同じポケモンでもステータスに差が出ます。また、個体値が 31 であることを V（32 進数表記で 31）、30 であることを U（32 進数表記で 30）と呼びます。

さらに、個体値はタマゴによって親から子へ遺伝させることが可能です。しかし、確実に遺伝させられるというわけではなく、両親の個体値が三つのステータスに遺伝し、残りはランダムに遺伝します。これを利用することで高個体値のポケモンを粘る事が出来ます。これが、ポケモンが廃人ゲーである一番の要因で、廃人達は高個体値のポケモンを入手するべく日夜、厳選作業[*13] に明け暮れるのです。

---

*10　マルマイン：別名「ばくだんボール」。最も書きやすいポケモンの内の一匹。ポケモン界最速だったのも今や遠い昔の話。

*11　カビゴン：実は、今回の記事で厳選する予定だったポケモンだったりする。

*12　各ステータス：具体的には、HP、こうげき、ぼうぎょ、とくこう、とくぼう、すばやさ、の六つを指し、それぞれを H（Hit Point）、A（Attack）、B（Block）、C（Concentration）、D（Defense）、S（Speed）と略します。

*13　厳選作業：高個体値のポケモンを選ぶこと。大抵は育て屋の前に意図的に用意されたであろう直線道路をタマゴを孵化させるためにウロウロすることが多い。DS の十字キーを上下させるだけの簡単なお仕事。

■実際に厳選してみよう

それでは実際に厳選してみましょう。今回の冬休みは大半がこの記事用の厳選作業を行う時間となりました。

まず、どのようなポケモンを厳選するか方針を決めます。

・どのポケモンを厳選するか。
今回はワンリキー[*14]を厳選します。

・性格[*15]をどうするか。
ようき[*16]（S ↑ C ↓）

・特性[*17]をどうするか。
ワンリキーは、こんじょうとノーガードの二種類。今回はどっちでもいいとする。

・タマゴ技[*18]はどうするか。
先制技のバレットパンチ（以降、バレパンと略す）を遺伝させたい。経路としては、エビワラー[*19]からアサナン[*20]を経由してワンリキーに遺伝させる。その際の三色パンチ[*21]が遺伝可能なのでついでに遺伝させる。

・個体値はどれでVを狙うか。
HASV[*22]は狙いたい。HAVで妥協も可能。

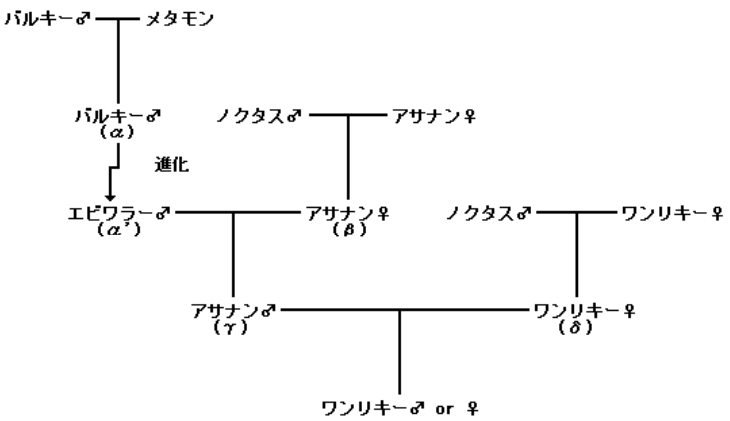
今回の方針をまとめると、『バレパンと三色パンチを遺伝させたワンリキーで、ようきかつHASV以上の個体が生まれるまで粘る』となります。次に、この方針を達成するための経路をまとめます。

---

*14　ワンリキー：かいりきポケモン。ゴーリキーを経て、最終的にはカイリキーになる。
*15　性格：全25種類存在する。性格によって成長しやすい能力があったり、成長しにくい能力があったりする。
*16　ようき：すばやさが上がりやすく、とくこうが上がりにくい性格。
*17　特性：ポケモンによって異なる特殊能力のようなもの。一つないし二つ持っている。
*18　タマゴ技：タマゴから生まれてきた時点で覚えている技。レベルアップでは覚えられない技が多い。♂がその技を覚えていることが条件。
*19　エビワラー：パンチポケモン。多くのパンチ技を覚える。♂しかいない。
*20　アサナン：めいそうポケモン。性別が片方しかない場合だと高個体値の親を厳選するのが面倒なので、ここを経由する。
*21　三色パンチ：ほのおのパンチ、かみなりパンチ、れいとうパンチの三つのパンチ技のことを指す。
*22　HASV：HPとこうげきとすばやさの個体値が31であること。

▼今回の経路

バルキー[*23] をからてだいおう[*24] からもらう。①
↓
メタモン(ようき) × バルキー♂
でバルキー♂(ようき)を入手。②
↓
上記のバルキーを育てて、エビワラー♂(ようき / バレパン、三色パンチ)を入手。③
↓
ノクタス[*25] ♂ (HASV) × アサナン♀
でアサナン♀(HAV or ASV or HSV or HASV)を入手。
同様にワンリキー♀(HAV or ASV or HSV or HASV)を入手。④
↓
エビワラー♂(ようき / バレパン 、三色パンチ) × アサナン♀(HAV or ASV or HSV or HASV)
でアサナン♂(ようき / バレパン、三色パンチ/HAV or ASV or HSV or HASV)を入手。⑤
↓
アサナン♂(ようき / バレパン、三色パンチ / HAV or ASV or HSV or HASV) ×
ワンリキー♀(HAV or ASV or HSV or HASV)
でワンリキー♂ or ♀(ようき / バレパン、三色パンチ / HAV or HASV)を入手。⑥
↓
厳選完了。



よくわかる経路図

---

*23 バルキー：けんかポケモン。レベル 20 のときのステータスが『こうげき＜ぼうぎょ』の場合、エビワラーに進化する。

*24 からてだいおう：格闘タイプのポケモンを主に扱う敵トレーナー『からておう』の親玉ではないかと囁かれている男。

*25 ノクタス：かかしぐさポケモン。タマゴグループがワンリキー、アサナンと同じ人型。別の厳選の時に入手。

●経路①　厳選に必要なポケモンを用意。
　スリバチやま[*26] 最深部へ向かう。からてだいおうの前で、『バルキーをもらう→ステータスチェック→リセット』を繰り返すこと数十回。バルキー♂（DV）を入手。

●経路②　親となるポケモンを探す。その１。
　ここでは、性格及びタマゴ技を遺伝可能な親（γ）に性格及びタマゴ技を遺伝させる親（α）を厳選します。

　『メタモン@[*27] かわらずのいし[*28](ようき）× バルキー♂@パワーバンド[*29]（DV)』を育て屋に預けるもタマゴができる気配なし。

しまった、バルキーはタマゴ未発見グループだ！！

　せっせとバルキーのレベルを 20 まで上げてエビワラーに進化させ、『メタモン@かわらずのいし(ようき）× エビワラー♂@パワーバンド』を育て屋に預ける。

　上下運動を繰り返すこと約 2 分、育て屋のお爺さんが横を向きました。これがタマゴができた合図です。話しかけて受け取りましょう。ちなみにこのお爺さん、タマゴができると主人公に電話で連絡してくれます。

　開始 8 分、タマゴが四つできました。かわらずのいしによる性格遺伝確率は 50 %、パワーバンドによる特防個体値遺伝確率は 100 %なのでこのうちの約二つはようき DV バルキーの確率が高いはずです。

　さあ、孵化歩数[*30] を半減する特性『ほのおのからだ』持ちのブーバー[*31] を手持ちに加え、後は自転車で走り続けるだけです。

---

*26　スリバチやま：ぞう k（ry もといマリルの生息地。稀にマリルが大量発生することがある。
*27　@：『@“持ち物”』と表記して、そのポケモンがどのアイテムを持っているかを示す。
*28　かわらずのいし：片方の親に持てせることによって親の性格を 50%の確率で子に遺伝させることのできるアイテム。
*29　パワーバンド：片方の親に持たせることによって親の個体値を一つだけ 100%の確率で遺伝させることのできるパワー系アイテムの一つ。パワーバンドはとくぼうの個体値を遺伝させることができる。
*30　　孵化歩数：ポケモンのタマゴは、タマゴを手持ちに加えた状態で一定の歩数を稼ぐと孵る。この歩数は種族ごとに異なっている。
*31　ブーバー：ひふきポケモン。三色パンチの内のほのおのパンチは元々このポケモンの専用技だった。孵化作業のお供。

孵化のための自転車上下運動開始から 7 分、四つのタマゴがすべて孵りました。

#1-1　バルキー / ♂ / おとなしい / DV
#1-2　バルキー / ♂ / ようき / DV
#1-3　バルキー / ♂ / ようき / DV
#1-4　バルキー / ♂ / ようき / DV

もしかしたら、2V[*32] がいるかもしれないのでバトルタワーの研究員[*33] で判定

#1-1　DV / まずまず[*34]
#1-2　DV / 平均以上[*35]
#1-3　DV / まずまず
#1-4　DV / 平均以上

判定の結果、2V はいなかった。とりあえず、この中で一番合計個体値が高そうな#1-2 のバルキーを選択。このバルキーをエビワラーに進化させて、性格及びタマゴ技遺伝可能な親に性格及びタマゴ技遺伝させる親とします。

●経路③　親となるポケモンを探す。その２。
ここでは、性格及びタマゴ技を遺伝可能な親（γ）に、性格及びタマゴ技を遺伝させる親（α）に技を覚えさせます。帰りの関鉄バスの中で育て屋などを駆使してレベル上げを行いました。

バルキー♂（α / ようき）→ エビワラー♂（α' / ようき/バレパン、三色パンチ）

●経路④　親となるポケモンを探す。その３。
ここでは、性格及びタマゴ技を遺伝可能な親の親（β）と、欲しい種族の親（δ）を厳選します。狙うのは 2V 以上のアサナン♀。

『ノクタス♂@パワーリスト（HASV）× アサナン♀（DV / 草むらで捕獲）』を育て屋に預ける。

開始から 20 分、二つのタマゴが孵る。

#2-1　アサナン / ♂ / いじっぱり / ASV

---

*32　2V：個体値 31 のステータスが二つあること。
*33　バトルタワーの研究員：大まかな個体値判定をしてくれる。一番個体値の高いステータスと大まかな個体値の合計を教えてくれます。一番個体値の高いステータスが複数ある場合は複数回聞くと教えてもらえます。
*34　まずまず：研究員が合計個体値が 90 以下であると判定したときに表示されるメッセージ『まずまずののうりょくをもっている』の略。
*35　平均以上：研究員が合計個体値が 91 ～ 120 であると判定したときに表示されるメッセージ『へいきんいじょうののうりょくをもっている』の略。

#2-2　アサナン / ♂ / まじめ / ADV

開始から 30 分。三つ目のタマゴが孵る。

#2-3　アサナン / ♂ / ずぶとい / AV

タマゴのできやすさが『親の種族が不一致＜親の種族が一致』であるがゆえ、このままでは効率が良くないと判断、#2-1 のアサナンをノクタスと入れ替えて、育て屋に『アサナン♂@パワーリスト[*36](ASV) × アサナン♀(草むらで捕獲 / DV)』を預け直す。

引き続き、上下運動。開始から 38 分、五つのタマゴを孵らせる。

#2-4　アサナン / ♂ / さみしがり / ASV
#2-5　アサナン / ♂ / しんちょう / ADV
#2-6　アサナン / ♂ / AV / DNDN[*37]
#2-7　アサナン / ♀ / ひかえめ / AV
#2-8　アサナン / ♂ /がんばりや / ASV

開始から 51 分、最初から預けていたアサナン♀（草むらで捕獲 / DV）を#2-7 のアサナンと入れ替えて、育て屋に『アサナン♂@パワーアンクル[*38](ASV) × アサナン♀(AV)』を預け直す。その後、さらに五つのタマゴを孵らせる。

#2-9　アサナン / ♀ / のんき / ASV
#2-10　アサナン / ♀ / むじゃき / ASV
#2-11　アサナン / ♂ / SV / DNDN
#2-12　アサナン / ♂ / SV / DNDN
#2-13　アサナン / ♂ / のうてんき / ASV

開始から 86 分、#2-9 のアサナンが条件を満たしたので、性格及びタマゴ技を遺伝可能な親の親（β）厳選を完了とする。

*36　パワーリスト：こうげきの個体値を遺伝させるパワー系アイテム。
*37　DNDN：ドナドナ。ようは逃がしたということ。
*38　パワーアンクル：すばやさの個体値を遺伝させるパワー系アイテム。

続いて、ワンリキーの厳選を開始。この厳選を行ったのは 12 月 25 日で孵化歩数減少日[*39] でした。狙うのは 2V 以上かつ、その 2V がアサナンと被らないワンリキー♀。

育て屋に『ノクタス♂@パワーリスト × ワンリキー♀（洞窟で捕獲)』を預けて、上下運動を開始。

開始から 27 分後、五つのタマゴが孵る。

#3-1 ワンリキー / ♀ / ASV / 相当優秀[*40]
#3-2 ワンリキー / ♀ / AV / DNDN
#3-3 ワンリキー / ♂ / AV / DNDN
#3-4 ワンリキー / ♂ / AV / 相当優秀
#3-5 ワンリキー / ♂ / ASV

ここで、最初に預けたワンリキー♀（洞窟で捕獲）と#3-1 のワンリキーを入れ替えて、育て屋に『ノクタス♂@パワーウエイト[*41]（HASV） × ワンリキー♀（ASV)』を預け直す。その後、更に五つのタマゴを孵らせる。

#3-6 ワンリキー / ♂ / HV / DNDN
#3-7 ワンリキー / ♀ / ゆうかん / HAV
#3-8 ワンリキー / ♂ / のんき / HAV
#3-9 ワンリキー / ♂ / HSV
#3-10 ワンリキー / ♂ / HAV

開始から 46 分、#3-7 のワンリキーが条件を満たしたので、欲しい種族の親（δ）の厳選を完了とする。

●経路⑤ 親となるポケモンを探す。その４
引き続き、性格及びタマゴ技を遺伝可能な親（γ）を厳選します。この厳選も孵化歩数減少日に行いました。狙うのは、ようき ASV のアサナン♂。

育て屋に『エビワラー♂@かわらずのいし(DV / ようき / バレパン、三色パンチ） × アサナン♀@パワーアンクル（ASV)』を預けて、上下運動を開始。

*39 孵化歩数減少日：タマゴが孵化するのにかかる歩数が減少する日。12 月 25 日以外には、1 月 12 日、2 月 14 日、4 月 15 日、5 月 5 日、6 月 11 日、7 月 7 日、8 月 21 日が孵化歩数減少日ではないかといわれている。
*40 相当優秀：研究員が合計個体値が 121 ～ 150 であると判定したときに表示されるメッセージ『そうとうゆうしゅうなのうりょくをもっている』の略。
*41 パワーウエイト：HP の個体値を遺伝させるパワー系アイテム。

開始から 36 分、六つのタマゴが孵る。

#4-1　アサナン / ♂ / ようき / SV / DNDN
#4-2　アサナン / ♂ / ようき / SV / DNDN
#4-3　アサナン / ♂ / ようき / SV
#4-4　アサナン / ♀ / ようき / DSV
#4-5　アサナン / ♀ / ようき / DSV
#4-6　アサナン / ♀ / なまいき / SV / DNDN

ここで、最初に預けたアサナンと#4-3 のアサナンを入れ替えて、育て屋に『アサナン♂@かわらずのいし(ようき/SV/バレパン、三色パンチ) × アサナン♀@パワーリスト (ASV)』を預け直す。その後、更に十一個のタマゴを孵らせる。

#4-7　アサナン / ♂ / ようき / AV / DNDN
#4-8　アサナン / ♀ / ようき / ASV
#4-9　アサナン / ♀ / ようき / ASV / DNDN
#4-10　アサナン / ♀ / ようき / ASV
#4-11　アサナン / ♀ / いじっぱり / AV
#4-12　アサナン / ♀ / てれや / ASV / DNDN
#4-13　アサナン / ♂ / がんばりや / ASV / DNDN
#4-14　アサナン / ♂ / おだやか / AV / DNDN
#4-15　アサナン / ♀ / おだやか / ABSV
#4-16　アサナン / ♂ / ようき / ASV
#4-17　アサナン / ♀ / ようき / ASV
#4-18　アサナン / ♂ / ようき / AV / DNDN

開始から 88 分、#4-16 のアサナンが条件を満たしたので、性格及びタマゴ技遺伝可能な親（γ）の厳選を完了とする。

●経路⑥　本厳選

ここでの厳選が終われば、厳選作業は終了です。ここで狙うのは、ようきワンリキー HASV、性別問わず、特性問わず。また、2V 以上の個体が生まれた場合に限り、ふしぎなあめ[*42] を使った UV 判定[*43] を行います。ワンリキーはレベル 10 で UV 判定可能です。

育て屋に、『アサナン♂@かわらずのいし(ようき / ASV / バレパン、三色パンチ) × ワンリキー♀@パワーウエイト (HAV)』を預けて、最後の上下運動を開始。

---

*42　ふしぎなあめ：ポケモンに食べさせるとレベルが 1 上がるアイテム。何味かは不明。努力値（育成編にて解説。ステータスの経験値のようなもの。）を稼がずにレベルを上げられるので便利。

*43　UV 判定：一定のレベルでのステータスの値を見ることで大まかな個体値を判定すること。U か V であることを判定することが多い。

開始から 43 分、八つのタマゴが孵る。ここでいったん中断する。

#5-1　ワンリキー / ♀ / ノーガード / やんちゃ / HV / DNDN
#5-2　ワンリキー / ♂ / ノーガード / ようき / HAV
#5-3　ワンリキー / ♂ / こんじょう / ようき / HV / DNDN
#5-4　ワンリキー / ♂ / ノーガード / ようき / HV / DNDN
#5-5　ワンリキー / ♂ / ノーガード / わんぱく / HV / DNDN
#5-6　ワンリキー / ♀ / こんじょう / ようき / HAV
#5-7　ワンリキー / ♂ / ノーガード / むじゃき / HASV / 素晴らしい[*44]
#5-8　ワンリキー / ♂ / ノーガード / ようき / HV / DNDN

再開から 48 分、九つのタマゴが孵る。集中力が持たないのでまた中断。

#5-9　ワンリキー / ♂ / こんじょう / おっとり / HV / DNDN
#5-10　ワンリキー / ♀ / ノーガード / ようき / HBV
#5-11　ワンリキー / ♀ / ノーガード / ようき / HAV / DNDN
#5-12　ワンリキー / ♂ / こんじょう / ようき / HAV / 相当優秀
#5-13　ワンリキー / ♂ / ノーガード / のうてんき / HAV / DNDN
#5-14　ワンリキー /♂ / ノーガード / なまいき / HBSV / 相当優秀
#5-15　ワンリキー / ♂ / こんじょう / ようき / HAV
#5-16　ワンリキー / ♂ / ノーガード / ようき / HAV
#5-17　ワンリキー / ♂ / ノーガード / いじっぱり / HV / DNDN

再々開から 53 分、八つのタマゴが孵る。UV 判定の結果、中々良いワンリキーが生まれたので厳選を終了とする。

#5-18　ワンリキー / ♀ / こんじょう / ようき / HAV / 相当優秀
#5-19　ワンリキー / ♂ / ノーガード / うっかりや / HAV / 素晴らしい
#5-20　ワンリキー / ♂ / ノーガード / せっかち / HSV /
#5-21　ワンリキー / ♂ / ノーガード / せっかち / HV / DNDN
#5-22　ワンリキー / ♂ / ノーガード / ようき / HAV
#5-23　ワンリキー / ♂ / ノーガード / のうてんき / HASV / 素晴らしい
#5-24　ワンリキー / ♂ / ノーガード / むじゃき / HV/DNDN
#5-25　ワンリキー / ♀ / こんじょう / むじゃき / HAV / DNDN

*44　素晴らしい：研究員が合計個体値が 151 以上であると判定したときに表示されるメッセージ『すばらしいのうりょくをもっている』の略。

■最終厳選結果

・#5-7　ワンリキー♂ / ノーガード / むじゃき / HASV
/『H / 31 , A / 31 , B / 20~24 , C / 20~21 , D / 20~21 , S / 31』

求める性格ではなかったものの、満足のいく個体値。むじゃきは S ↑ D ↓なので、S ↑ C ↓のようきと運用の仕方が少し違うだけなので採用。

・#5-12　ワンリキー♂ / こんじょう / ようき / HAV
/『H /31 , A / 31 , B / 0~9 , C / 30 , D / 30 , S / 0~9』

なぁにこれ。ほとんど 4V じゃん。性格がいじっぱりだったら最高じゃん。なんでようきで粘ってるときにこうなるかね。親として使えるからまだいいけどさ。ちょっと待てよ……ネタ型[*45] にすればいいじゃない。

・#5-18　ワンリキー♀ / こんじょう / ようき / HAV
/『H / 31 , A / 31 , B / 20~29 , C / 10~19 , D / 0~4 , S / 30』

今回の狙い通り。D の低さが気になりますが、それ以外は文句なし。欲を出せば、C と D が逆ならさらによかった。採用。

■あとがき

今回で、最も時間がかかる厳選作業は終了しました。孵化させたタマゴの数は 70 個、厳選作業にかかった時間は約 12 時間でした。性別を限定したり、めざめるパワー[*46] を粘ったりしていないので、割と軽めの厳選作業だったりします。

さて、次は楽しい楽しい**育成**です。ガチ型のカイリキー[*47] とネタ型カイリキーを合わせて三体ほど育成する予定です。しかしこのままでは腕が 12 本で暑苦しいので、清涼感のある別のポケモンも平行して育成したいと思います。

次回、

ポケットモンスター廃人日記～育成編～（仮）

ご期待ください。

---

*45　ネタ型：実用性を求めず、ネタに走る育て方。『高個体値のポケモンでやるほどネタネタしくて良い』というのが私の持論。

*46　めざめるパワー：ポケモンの技の一つ。ポケモン一体ごとにタイプと威力が異なるため、狙ったタイプ、威力を粘り出すときりがない。

*47　カイリキー：かいりきポケモン。ワンリキーの最終進化型。腕が 4 本あり、いろいろと逞しい。

たたかうかえる

闘蛙

文： 編集部 ふぁい

■あいさつ

こんなくだらない記事しか書いてないのに、なんか編集長になってました。

それはともかく、今回攻略する 4 面から**難易度が急上昇**するので、**記事のテンションも急上昇**するかもしれません。

■ Stage4:*かえるエクスプレス*

この面は、強制横スクロール面です。勝手に前に進むバイクに乗り込み、次々と現れる障害物を避けて進むだけの面です。面の紹介も兼ねて、実際にプレイしてみるとどのようになるか、紙上体験プレイをご覧いただきましょう。

さっそうとバイクに乗り込む主人公（カエル）。乗り込むや否や、バイクは勝手に走り始めます。それにしても目つきの悪い主人公（カエル）だ。

ちなみにこのゲームは、各ステージごとの操作方法などの説明が**一切無い**ので、本能のおもむくままにボタンを操作して、操作方法を覚えるしかありません。

3 秒ほどすると、前方に点滅している塀が表示されます。

これが何を意味するかの説明もまったくありませんが、ここまで来たプレイヤーならレア社の意地の悪さを何度も目の当たりにしているため、**一抹の不安**を感じるはずです。

そして次の瞬間……

塀が突然現れてぶつかりました。ひしゃげるバイク。衝撃ですっとぶ主人公(カエル)。スタートしてから**約 5 秒**で**即死**しました。

勘の良い方なら気づくかもしれませんが、先ほどの塀の点滅表示は、「いまから塀が出現するよ」という**予告**だったのです。

ちなみに点滅表示から実際に塀が現れるまでは**約 2 秒**しかブランクがありませんが、これでもレア社的には**親切なチュートリアル**のつもりらしく、少しすると点滅表示から障害物が現れるまでのブランクが**約 1 秒**ほどになります。

障害物に当たって死ぬと残機が減って、スタート地点、または中間地点(後述)からのやりなおしとなります。

Q,あれっ？じゃあ HP って何の意味があるの？
A,**何の意味も無いです。**

Q,なんか点数が増えてるんだけど、何で？
A,走行距離に応じて点数が増えます。
ただ、増えるだけで**何の意味も無いです。**

右の方に見えるバーが中間地点です。ステージ内に数カ所設置されています。

ここを通過すれば、死んでもここからやり直しできます。

しかし残機が尽きてコンティニューすると、容赦なく**スタート地点からのやり直し**となります。

さらにコンティニューには回数制限があり、5 回でコンティニューが切れます。

そして……

・゜★.・゜ ゜☆.・・゜☆.・゜ ゜★・*・

GAME　OVER

・゜★.・゜ ゜☆.・・゜☆.・゜ ゜★・*・

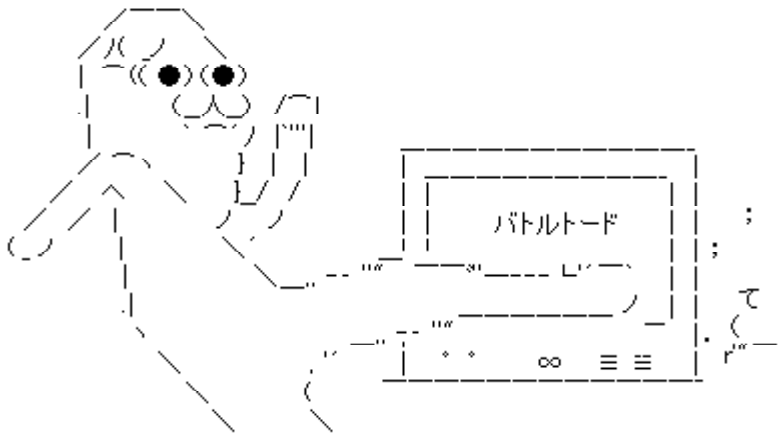
「わたしを　たおすには
まだまだ　しゅぎょうが　ひつようね。
ハーッハッハッハ！」

オープニングをすっとばすと「誰やねん！」と思ってしまいますが、この~~ババア~~お姉さんは悪役の「ダーククイーン」です。ラスボスです。ドラクエでいえば竜王です。

こちらは味方のＴバード博士。
「コンティニューが　きれた！
もういちど　**さいしょから**じゃ！」

えっ！？　さいしょから……？

オープニングに戻された。
ほんとうに　さいしょからじゃ！！１１！



以上が紙上体験プレイです。この面の操作自体は単純なのですが、**障害物の予告が 1 秒にも満たなかったり**、**2 ～ 3 個の障害物の予告がいっぺんに出てきたり**など、**エスパークラス**の判断力が要求されるため、難易度の高い面に仕上がっています。

初めてプレイする人は、大体この面でコンティニューが尽きて 1 面からやり直しとなり、深い悲しみと憤りのあまりにコントローラーを投げ出してしまいます。

紙面上では、ステージの~~グロテスク~~**ハイセンスな配色**と、**やたらカッコイイ BGM** が伝えられないのが残念ですが、このあたりでステージの攻略にいきたいと思います。

■攻略のコツ

前述の通り、初見でクリアするにはエスパークラスの判断力が要求されます。むしろエスパーじゃないとクリア出来ないかもしれません。

そんなわけで我々一般人に残された方法はただ一つ。そう、それは**コースを覚える**こと。実際に自力でクリアしようとすると、コンティニューにコンティニューを重ねるうちに、自然にコースを覚えてしまいます[*1]。

*1 そして勉強時間が削られて単位を落とします。

■操作説明

操作は至って単純です。
上下左右キーで移動が出来るほか、A、B、X、Y ボタンでジャンプできます。

■障害物一覧

◆塀
道幅いっぱいに広がる塀です。
ジャンプしないとぶつかって死にます。
ジャンプで避けましょう。

◆浮遊塀
空中に浮かぶ塀です。
通常の塀と違って、ジャンプするとぶつかって死にます。
通常の塀と間違えてジャンプしないように注意しましょう。

◆石板
道幅の半分の幅をもつ大きな石板です。
ジャンプしてもぶつかって死にます。
上下に移動して避けましょう。

◆ジャンプ台
これに乗ると、通常のジャンプよりも大きく飛べます。
大抵はこれに乗らないと穴に落ちて死ぬので乗りましょう。

◆空中ジャンプ台
空中に浮かんでいるジャンプ台です。
ジャンプしながら乗らないと大ジャンプできません。

◆二重ジャンプ台
ジャンプしながら乗ると、通常のジャンプ台よりもさらに大きなジャンプができます。

◆ラットポット
この面で唯一出てくる敵です。倒すことは出来ません。
触れると死にます。

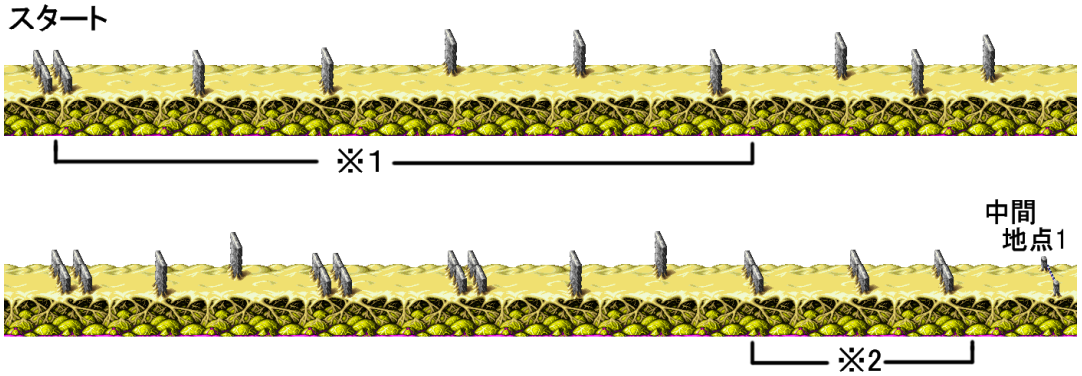
.

闘蛙

■ MAP

1 面、2 面のように、MAP ＋注釈の形で攻略をします。
ただし誌面の都合上、障害物と障害物の間の距離をかなり縮めて描いてあります。
ＭＡＰは上から下に並んでいます。

◆スタートから中間地点 1 まで



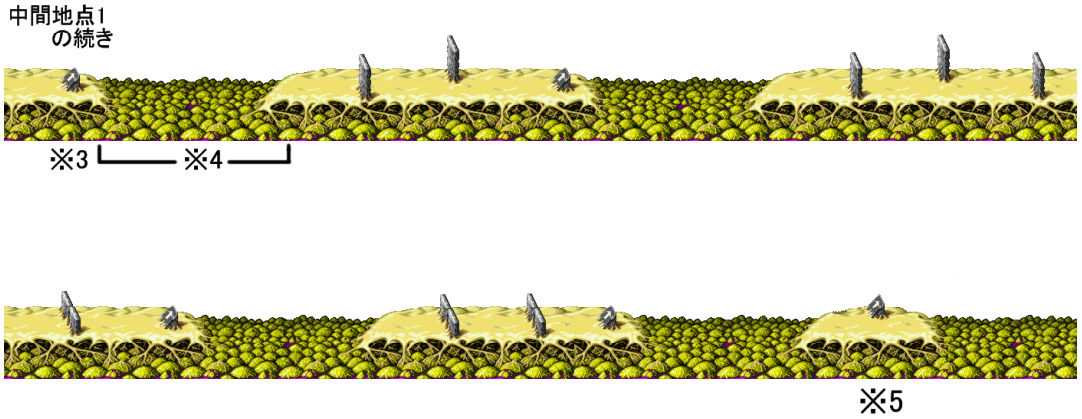
※ 1

このあたりは障害物の予告から現れるまでの時間や、障害物同士の間隔が長めにとられています。この後と比べて障害物を避けるタイミングが違うため、コンティニュー切れでここまで戻されたときには注意しましょう。

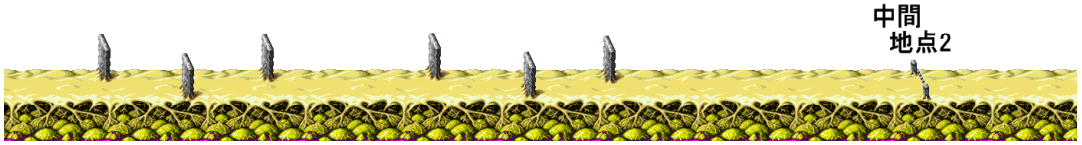
※ 2

塀が 3 連続で現れます。塀と塀の間隔はかなり短いので、ジャンプボタンを連打してもいいかもしれません。

◆中間地点 2 から中間地点 3 まで



※6



※ 3

ここで初めてジャンプ台が登場します。ジャンプ台の幅は道幅の 1/3 ほどなので、主人公の位置を合わせないとジャンプ台に乗れません。

※ 4

穴です。腐ったレモンのようなものがびっしりと敷き詰められています。ここに落ちると死亡扱いとなります。

※ 5

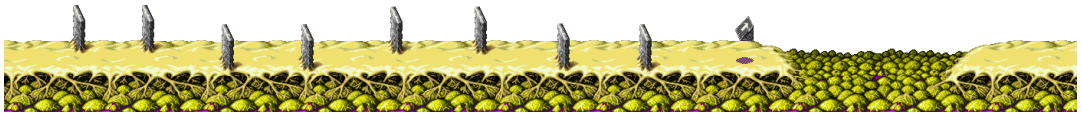
コースの上側(奥側)にジャンプ台が設置されています。上(奥)に移動しないとジャンプ台に乗れずに穴に落ちて死にます。

※ 6

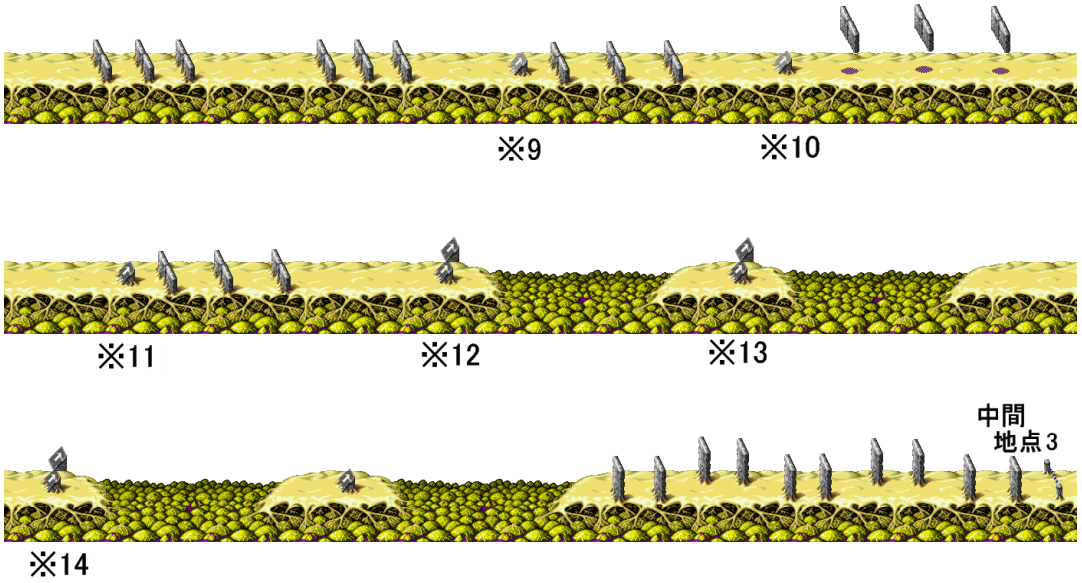
今度は下(手前)にジャンプ台が設置されています。

◆中間地点 2 から中間地点 3 まで

※7



※8



※ 7

ここで初めて浮遊塀が登場します。何も操作しないでくぐれば良いだけなのですが、通常の塀とまちがえてジャンプしてしまわないように注意してください。

※ 8

ここで初めて空中ジャンプ台が登場します。ジャンプしてのらないとジャンプ台の効果は得られません。乗るためにジャンプするタイミングに注意してください。

※ 9

コースの真ん中にあるジャンプ台に乗らないと、3 つ連続した塀を越えることはできません。

※ 10

※ 9 の逆で、ジャンプ台に乗ると浮遊塀に当たって死にます。コースの奥か手前に避けましょう。

※ 11

※ 9 と同じくジャンプ台に乗らないと死にます。

※ 12

ここで初めて二重ジャンプ台が登場します。ジャンプせずに乗ると通常のジャンプ台と同じ効果が、ジャンプしながら乗ると通常のジャンプ台の 2 倍ほどのジャンプ力が得られます。

ここではジャンプしながら二重ジャンプ台に乗っても、ジャンプせずに乗っても画面の右端によれば先に進めます。

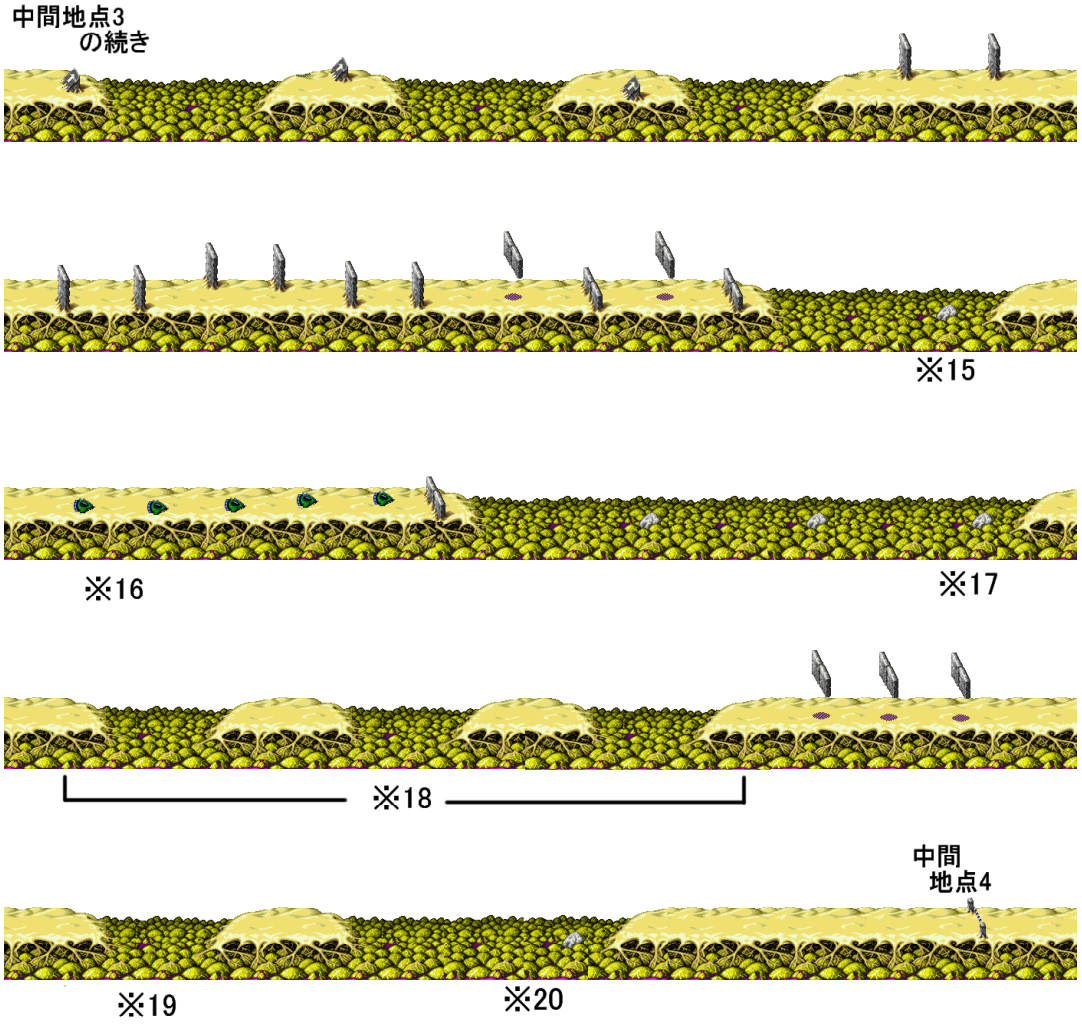
※ 13

ジャンプしながら乗らないと穴に落ちて死にます。

※ 14

ここではジャンプしながら二重ジャンプ台に乗っても、ジャンプせずに乗っても画面の右端によれば先に進めます。

◆中間地点 3 から中間地点 4



※ 15

ここはどうあがいても穴に落ちてしまいますが、レモンのようなものが敷き詰められている中に石があり、そこでジャンプボタンを押せば穴から脱出できます。

石のところでタイミングよくジャンプボタンを押す必要はなく、石が見えたらジャンプボタンを連打すれば大丈夫です。

※ 16

ラットポットが道の奥と手前を往復しています。避けるかジャンプして飛び越せば大丈夫ですが、わざわざそんなことをしなくても**画面の左下が安全地帯です**。左下にいれば当たることはありません。

※ 17

穴の中に石が 3 つありますが、3 つ目の石がみえたらジャンプボタンを連打すれば大丈夫です。

闘蛙

※ 18

穴が 3 つ連続しています。かなりジャンプのタイミングがシビアなので注意してください。

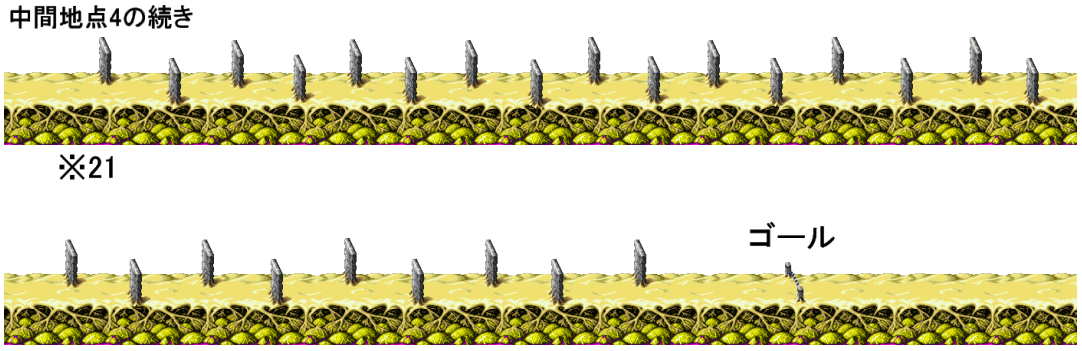
※ 19

油断してるとこの穴の存在を忘れがちになります。注意してください。

※ 20

ここにも石があります。※ 15 と同じようにジャンプしましょう。

◆中間地点 4 からゴール



※ 21

ここからがレア社の本気です。

石板が奥と手前に交互に設置してあるだけなのですが、なんとここから障害物の予告がなくなります。しかも**バイクの速度がこれまでの 2 倍くらいになります。**

まさに**悪魔の仕業**です。**レア社に悪魔が入社したに違いありません。**

タイミングよく十字キーの上と下を押して、切り抜けましょう。ボタンを押すタイミングはかなりシビアです。アクションゲームではなく音ゲーと割り切ってボタンを上下に押しましょう。

■おわりに

なんと今回攻略した 4 面は鬼畜面の序章に過ぎません。テニスの王子様でいえば、週間少年ジャンプ連載時みたいなものです。

次回はレア社の**選び抜かれた悪魔精鋭が作り上げた 5 面**の攻略をします。お楽しみに。

GRな日々。II ～さよなら能登/北陸号～

文 編集部 葡萄酒

夜行の浪漫

私の実家は富山にある。読者の皆は「北陸」と聞いて、たとえば東京から金沢への経路が想像できるだろうか。これは地元民以外には、なかなか難しいと思う。答えを言ってしまうと、現実的なルートとしては新潟回り(新幹線から越後湯沢で北陸本線に連絡)が一般的である。しかし日本地図を見てもらえば分かるのだが、このルートは実に遠回りである。日本アルプスが本州の中央に鎮座しているせいで、首都圏から直線的に移動することが出来ないのだ。加えて北陸本線の所要時間がネックになる。こう書くと北陸本線はトロトロ走っているように聞こえるかもしれないが、長い駅間距離のおかげで速度に関しては在来線最速と名高い特急「はくたか」など、トップクラスなのだ。しかし、いかんせん距離が距離である。たとえば金沢-越後湯沢間の 260km 弱、最高時速 160km を誇る「はくたか」を駆使しても 2 時間 40 分ほどを要し、乗車券と指定席特急券を併せて約 8,000 円かかる。更にここから上越新幹線に連絡し、1 時間弱の時間と 7,000 円近くかけて東京駅へとたどり着けるのである。所要時間は約 4 時間、運賃は計 15,000 円弱。これを頻繁に利用するのは貧乏大学生には難しい。

そこで、浪漫溢れる「夜行列車」の登場である。北陸から首都圏へ(無論下りに関しても同じ)と向かう夜行列車は急行「能登」号と寝台特急「北陸」号があり、どちらも運行区間は金沢-上野間となっている。

まずは急行「能登」号から見ていこう。この列車は、1959 年に開始された金沢-上野間を結ぶ急行「黒部(くろべ)[*10]」が元になっており、現在運行している数少ない「急行」の一つである[*2]。車両は 9 両編成の全車両が座席車両となっており、自由席と指定席[*3]、グリーン席がある。停車駅は上野駅 - 大宮駅 - 熊谷駅 - 高崎駅 - 直江津駅 - 糸魚川駅 - 泊駅 - 入善駅 - 黒部駅 - 魚津駅 - 滑川駅 - 富山駅 - 小杉駅 - 高岡駅 - 石動駅 - 津幡駅 - 金沢駅。私は乗り換えが苦手なので、一度も車両を降りずに目的地までたどり着けるのは嬉しい。上りは金沢を上野を 23:33 に発車、金沢に着くのが 6:35。寝ている間に長距離移動できるので、時間の節約になるのも魅力だ。

しかしなんといっても、この「能登」最大の強みは自由席である。指定席が取れそうにないラッシュ時でも(目的地まで立つ覚悟、もしくは数時間前からホームに並ぶ覚悟があるのなら)前もって切符を買っておく必要がない。私の経験上、3 時間以上前から並んでいる人はまずいないので、2 時間くらい前から並んでおけば余裕で窓側の席が取れる。ノート PC を広げるのであれば、一番後ろの席が他人に画面を見られないので良いだろう。逆に一番前の席は足下に余裕があるので、少しでも多く休みたい人はそちらのほうがオススメだ。座席は(簡易的ではあるが)リクライニング可能なので、慣れれば快適に寝ることが出来る。余談だが、数年前までは自由席車輌が喫煙車になっており、禁煙車である指定席車輌からマナーの悪い~~クズ共~~利用者が自由席までたばこ

---

*10 富山県北東部の地名。紅白歌合戦で中島みゆきが「地上の星」を歌う舞台として用いられた黒部川第四発電所・黒部ダム(通称:黒四ダム)が有名。

*2 現存する急行列車は、本文中の「能登」に加えて青森-札幌間を結ぶ「はまなす」、大阪-新潟を結ぶ「きたぐに」の 3 つのみである。

*3 1 号車は指定席女性専用車両になっている。

を吸いに来るために空気が白く曇っており、車両の一番奥まで見通せないという酷い状況であったが今は全車禁煙になっている。また、6 号車にはラウンジがあり、ソファなどが利用できる。ちなみに混雑時はソファまで埋まるので注意しよう。編集部 Flast 氏の報告では、ラウンジに 5 つ程あるコンセントのうち自販機が挿さっている所の空いてる側は電気が来ているらしい[*4]。余談だが、閑散期に時々ソファで寝ている乗客を見かけるのだが、読者諸君は絶対にしないでもらいたい。体調が悪いときは仕方がないのだろうが、ただ「楽に寝たい」というのなら遠慮するべきだ。ラウンジは寝る場所ではなく「談話する場所」である。誰かが寝ていると、起こしてしまいそうで会話を躊躇してしまう。

運賃は新幹線と比べて格安で、上野から金沢までは自由席で 9,240 円(乗車券+急行券)、なんと新幹線を使ったときの 66%である。指定席券も通常 510 円(閑散期は 310 円)と安いので、利用できるならそちらのほうが良いだろう。ただ閑散期に関しては乗客がほぼゼロの時もあるため、指定席券はおそらく不要。秋休みに諸事情で帰郷した際は最寄り駅である JR 高岡駅から「能登」に乗ったのだが、同じ車両に誰一人おらず、座席を回して向かいの席を荷物置きに、更に自分は横になって眠れるという夢のような電車旅であった。ちなみに、後に糸魚川あたりでもうひとりだけ同じ車両に乗客が来たが、ガラ空きの車両を前に人間考えることは皆同じであったようだ。

いくつか注意事項としては、混雑期に自由席を利用する場合は必ず始発駅から乗るということと、あらかじめ飲食物を用意しておくことの 2 点である。前者については説明は不要だろうから後者について補足しておこう。簡潔に言うと「能登」「北陸」ともに車内販売が存在しない。自販機が貧弱な上に空調が完璧とはとても言い難いので、最低でもペットボトルの飲み物は事前に購入しておいた方が良いだろう。更に付け加えて言うと、上野駅のホームは非常にわかりにくい場所にあるので注意しよう。目指すのは 16 番ホーム、在来線特急乗り換え口の奥である。自動改札で乗車券だけを通し、16 番ホーム入り口で駅員さんに急行券を見せよう。ホームの位置は中央改札(アトレ側)入って右手側に進み、奥に見える切符売り場の左側に入り口がある。

「北陸」の B ソロ(2 号車)廊下

次は寝台特急「北陸」だ。もともとこの列車は北陸経由で上野 - 大阪間を直結する急行列車であった。夜に上野を出発し朝に北陸を通過、その後は昼行列車として大阪へ向かっていた。今までに何回かの廃止・再編を経て、現在では「能登」とほぼ同じ経路を走行する夜行列車となっている。上野駅 - 大宮駅 -

*4 そのコンセントは客が勝手に使って良いのか知らないので、使う前には乗務員に聞いてみた方が良いかもしれない。

高崎駅 - 水上駅[*5] - 長岡駅 - 直江津駅[*6] - 糸魚川駅 - 魚津駅 - 富山駅 - 高岡駅 - 津幡駅 - 金沢駅
車両編成は開放式 B 寝台、B 寝台個室「ソロ」、A 寝台個室となっており、8 両編成の 2 号車にはシャワー室がある。恥ずかしながら私は「北陸」のシャワー室を利用したことはないのだが、(シャワーに関しては)ほぼ同型の寝台特急「サンライズ出雲」号のシャワールームを利用した時には予想以上の快適さに驚いたものである。シャワールームに行く前にシャワーカードを乗務員から購入[*7] しよう。これ一枚で 6 分間お湯を出すことができる。頭や体を洗うためにお湯を止めているときは時間にカウントされないのでご心配なく。揺れる足下が少し不安定ではあるが、間違いなく宿舎風呂よりは快適である。JR 特製のタオルセットを購入[*8] することも出来るので、記念に入手しておくのも良いかもしれない。

「サンライズ出雲」シャワールーム

さて、ようやくお楽しみの寝台である。またまた恥ずかしながら、私は A 寝台を利用したことがない。実に片手落ちであるが、本記事では開放式 B 寝台(以下 B 寝台)と B 寝台個室(以下 B ソロ)をメインで扱うことにしよう。B 寝台は、2 つの二段ベッドが向かい合わせに配置してあり、それぞれの寝台ごとにカーテンで外からは見えないようになっている（本当はこちらの写真も用意したかったのだが、他の乗客も隣接した寝台を利用している関係で写真が用意できなかった。申し訳ない)。実は着替えるときに寝台の明かりをつけていると、カーテンに自分のシルエットがうっすらと投影されてしまうという脆弱性があったりする。別に中が見えるというわけではないのだが気になる方、特に女性は注意した方が良いだろう。この B 寝台は空調が酷く、かなり暑い場合がほとんどであるため、暑いと寝られないという方は後述の B ソロを利用することをオススメする。また、二段になっているため頭上のスペースがほとんど無い。これに関しては通常の二段ベッドと似たようなもので、別に飛び起きたからといって頭を打つ程ではない。あと、荷物を置く場所が用意されているわけではないので、実際に使えるスペースは広くない。重い旅行鞄

*5 水上駅と長岡駅は方向反転などの運転停止のみ。
*6 上り列車のみ停車。
*7 310 円。A 寝台利用者には無料で配布される。
*8 300 円。

を持っていく予定があるのなら、緑の窓口で「下段のベッドがいい」と言えば手配して貰えるはずだ(空席があれば)。とまあ、ここまでぐちぐちと文句を言ってきたわけだが、はっきり言ってこの B 寝台は快適とは言い難い。なぜなら、この列車の真価は B ソロに集約されているからだ。

B ソロの部屋内部

一番最初に言っておこう。この B ソロは B 寝台と値段が同じである。もし空席があるようなら、迷わずこちらを利用した方が良い。実際の B ソロの設備だが、とりあえず写真を見ていただこう。個室自体は少し手狭で天井も低い[*9]。しかし、実際に寝台に座ってみると、これでも十分な広さであることが分かる。備品としてハンガーが 1 つ、JR のロゴがプリントされた浴衣と帯が 1 セット、枕、毛布を使うことが出来る。また、乗務員に切符のチェックをしてもらう際に個室のカードキーが渡され、トイレに行くときなどに部屋に鍵を掛けることが出来るようになる。このカードキーは使い捨てで降車するときに返却する必要もないため、これも記念に貰っておこう。

まずは個室に入ったら、ベッド脇の柵をたたんでしまおう。上の個室は結構揺れるため転倒防止には便利なのだが、寝るとき以外は邪魔になる。次に、個室の暖房は強めにかかっている場合が多いので、暑い場合は空調を切ってしまおう。また、空気も悪くなりがちなので換気扇を付けた方が良い。それなりに音はするが、列車自体の音の方が遙かに大きいので誰も気にしないだろう。ついでに通路側のカーテンも閉めてしまった方が良い。部屋の準備が出来たら、後は寝台を満喫するだけだ。寝台に腰掛け、買っておいたコーヒーでも飲みながら夜景を眺めよう。まさに浪漫、実に優雅なひとときである。冬ならば新潟付近で美しい雪景色が目に入るはずだ。

さて、流れる夜景を一通り満喫したところで、列車はトンネル続きの山道に入る。もし翌日に予定があるのならば、そろそろ就寝である。寝る前の準備として、荷物は纏めておいた方がよい。明朝は大宮着の 15 分前、5:40 頃から車内放送が始まるのだが、あらかじめ準備しておいた方が無難だろう。上り列車であれば、上野まで乗ることが多いので少しくらい起きるのが遅れても問題ないのだが、下りで寝過ごすと悲惨なことになってしまう。

気になる運賃だが、実は新幹線ルートと比べても大きく変わらない。金沢 - 上野間の運賃は乗車券、特急券、寝台券併せて 17,110 円である。A 寝台を利用する場合は、24,160 円になる。新幹線よりも若干高いが、首都圏での行動予定に余裕を持てることを考えると十分「アリ」だろう。しかもこの運賃、JR のトクトクきっぷを使うと、更に安くなるのだ。その名も、上りは「首都圏往復フリーきっぷ」、下りは「北陸フリーきっぷ」である。前者は金沢 - 東京都区内を 22,320 円

*9 B ソロの車両は、上下二段で個室が並んでいる。

で在来線特急+新幹線の指定席を使って往復できる切符なのだが、実はこの切符は「北陸」の B 寝台と B ソロも利用できるのだ。往復までの猶予は 7 日間、この期間内は制限なしで区間内[*10]を乗り放題のオマケが付くので、新幹線より圧倒的に安い！後者の「北陸フリー切符」もほぼ同様で、北陸地方の区間を制限なしで利用できることと、滞在日数が 4 日になっていること以外はほとんど同じである(価格は 21,400 円)。条件が合うなら、これを利用しない手はない。しかも、この状態で更に学割を適用することが出来る。通常の切符よりは 1 割と効果が薄くなっているが、そもそもトクトクきっぷ自体が既に割引された値段なので、お得であることに変わりはない。もちろん、前述の単体で「能登」「北陸」を利用する際に 2 割引の学割が適用できる。

乗り場は上野駅中央改札を出て一番左、13 番ホームである。「能登」号とは違うホームなので注意しよう。乗り換え口などは特に用意されていないので、特急・寝台券については乗車後に乗務員からハンコを貰おう。

と、今まで 2 つの夜行列車について紹介してきたのだが、ここで読者の皆に哀しいお知らせがある。サブタイトルで予想が付いた人もいるかもしれないが、この「北陸」「能登」の両列車は 3 月 12 日のダイヤ改正で定期運行を終了することになっているのだ。どうも閑散期の利用者の少なさと、前述のトクトクきっぷを用いると格安で利用できるため、採算が取れなくなったという理由らしい。今後の運行予定としては、週末と繁忙期に臨時便として運行する予定であるということである。夜行列車ファン、また常連の利用者としては非常に寂しいところではあるが、受け入れるほか無いだろう。ちなみに金沢駅では「能登」「北陸」が並んでホームに停車している姿を見ることができ、写真を撮りに来るファンも多いとか。

ほとんど写真も GRDII も関係なかった第 2 回「GR な日々。」であるが、お楽しみ頂けたであろうか。いま私がこの記事を書いている 1 月 8 日現在、あと運行終了までは 2 ヶ月弱ある。もし少しでも興味を持ってくれる読者がいたら、利用してみることをオススメして、この記事の締めとしよう。北陸は米も魚も旨い。大学生活に疲れた人は、気分転換に北陸へ向かい「キトキト[*11]」な魚を味わうのもオツなものではないだろうか。

*10 http://www.jreast.co.jp/tickets/info.aspx?GoodsCd=974

*11 富山弁で「生きの良い」「新鮮な」を表す。

情報科学類の緑化事業

文 編集部 Flast

!!WARNING!!

前回[*1]の記事を読み飛ばした方も、そうでない方もこんにちは Flast です。前回の最後にマンデルブロ集合をやるとか言いましたが、気が変わったのでちょっと別のことを書きます。多分ほんとに次回はマンデルブロ集合です。

ということで今回の内容は CUDA と全く関係なく、ラムダ計算とか関数型言語とかに興味のある方向けとなっています。そうでない場合、読み飛ばした方が堅実です。あと w や W や v といった文字に嫌な思い出がある方も、読み飛ばした方がいいでしょう。

また、本文中にラムダ計算[*2]や BNF 記法[*3]といったものが出現しますが、本記事ではそれらの詳細については解説しません。あくまで緑化事業について解説します。

hello,esolang!!

みなさんは Brainf*ck[*4*5]という言語をご存知でしょうか、esolang[*6]として名高いあれです。

また、Brainf*ck はその言語仕様故に数多くの派生言語を誕生させました。その派生言語の 1 つ Grass を今回紹介しようと思います。

ところで、Brainf*ck ではプログラムを表現する為に 8 文字[*7]用います。そして、たった 8 文字でもチューリング完全なプログラミング言語です。それに対し、Grass では wWv の 3 文字[*8*9]しか用いません。もちろんチューリング完全です。

その他にも様々な esolang があり、なかには記事にすると発禁になる可能性[*10]がある言語[*11]ま

---

*1Flast, 文明の進歩の尺度への道(1/4dt), WORD12 号 WORD は一太郎を使用しています号, 2009, p.- .とか参考文献風に書いてみる。

*2 細かい説明は面倒なので http://ja.wikipedia.org/wiki/ラムダ計算

*3 バッカス・ナウア記法。詳細は http://ja.wikipedia.org/wiki/BNF

*4http://ja.wikipedia.org/wiki/Brainfuck

*5 アスタリスクには u が入りますが、卑語を含むのでこのように伏せ字になることが多いです。

*6esolang: Esoteric programming language 難解プログラミング言語のこと。故意に可読性や理解容易性を低下させた言語。正直これをメインの言語にはしたくない。

*7 その 8 文字は ><+-.,[] となっているが、別にこの 8 文字である必要はないとされている。また、それ以外の文字はすべてコメントとして認識される。

*8 同じように 3 文字しか用いない言語として Whitespace(http://compsoc.dur.ac.uk/whitespace/)というのがあるが、こちらは 空白、タブ、改行 と通常のエディタでは不可視な文字を使う。

*9 文字数を究極的に減らした言語としては A というのが存在するが、これは A 一文字でプログラムを表現する。ちなみに A での Hello, world は A が 8 進数で 22222222260222222220222222222220222221113704022422222224422240343333333333334122222222433333333422243333334333333334024 個続く。

*10 テニスみたいなのをしているそこのあなた。その先にはペスカ星しかありません。気をつけてください。

*11Misa という言語ですが、これについては各自の責任で検索等を行ってください。

であります。

記事に出来ないのは仕方ないので、とりあえず次節で Grass の細かい言語仕様について説明します。

言語仕様

前述のとおり、Grass では wWv の 3 文字のみ[*1] を使い、それ以外の文字はすべてコメントとして扱われます。wWv が大量に羅列されているのが草に見えたことが、Grass という名前の由来[*2]です。

Grass は型無しラムダ計算が元となっているので基本的な構文は、関数定義（ラムダ抽象）と関数適用に分けられます。

また、現在の実行環境を保存するスタックがあり、Grass 抽象機械[*3] は実行と同時にスタックを初期化します。また、スタックは上から順に 1,2,3,...とインデックスがつけられます。

まず関数適用ですが、すべての関数適用は任意の数(>0)の W と 同じく任意の数(>0)の w から構成されます。BNF で表すと以下のようになります。

```
〈W〉 ::= "W" | "W" 〈W〉
〈w〉 ::= "w" | "w" 〈w〉
〈app〉 ::= 〈W〉〈w〉 | 〈app〉〈app〉
```

関数適用が続く場合は列挙することが出来ます。

最初の W の数で適用する関数を示し、次の w で関数に適用する引数を示します。それぞれの数がスタックでのインデックスと対応します。そのため、WWwwww という場合はスタックの 2 番目にある関数に 4 番目を適用するという意味になります。

お気づきとは思いますが、この方法だと複数の引数を与えることが出来ません。この場合は部分適用（カリー化）を行います。つまり関数と引数をさらにクロージングした新たな関数をスタックにプッシュします。

関数適用や部分適用の例をあげてみます。まずスタックの 3 番目に 2 引数を受け取る関数があったとして、この関数に 1,2 番目の値を適用するとします。

```
WWWwWwww
```

この場合、最初の部分適用でスタックにさらに関数がプッシュされるので、スタックのインデ

*1 厳密には半角と全角それぞれの wWv をプログラムとして認識するため 6 文字。

*2Grass は昔 VIP 語と呼ばれていた時期があったが、その由来はご想像の通りだろう。

*3Grass のインタプリタやコンパイラなど。

ックスが変更される点に注意してください。

この時スタックの 1 番目には関数の戻り値が、2 番目には部分適用された関数が、3,4 番目には関数に渡した引数が、そして 5 番目には関数があります。

上記コードはラムダ式で書くと以下の様になります。f が関数、x,y がそれぞれ引数です。

```
fxy
```

次に関数定義ですが、全ての関数定義は任意の数(>0)の w から始まり、任意の数(>=0)の関数適用がそれに続きます。また、Grass のコードは w から始まり、それまでに出現したいかなる文字もコメントとして扱います。

BNF では関数定義は以下のようになります。

```
〈def〉 ::= 〈w〉 | 〈w〉〈app〉
```

つまり何もしない関数を定義することが可能ということです。関数定義の場合は関数適用と違い、羅列して書くことが出来ません。関数定義は必ず v で区切る必要があります。

関数が定義されると、定義された時点のスタックがクロージングされ、その関数が定義した時点でのスタックへとプッシュされます。また、関数適用される（関数呼び出しされる）とクロージングしたスタックへ引数がプッシュされ、関数適用されている間はそのスタックに束縛されます。

関数の戻り値は、最後に実行された関数の戻り値が返され、値自体は関数を適用した関数のスタックへとプッシュされます。

例として、3 引数を受け取り何もしない関数と、2 引数を受け取り何かする関数を定義すると以下のようになります。

```
wwwvwwWWWwwWwwWwww
```

引数を受けとるだけの関数は、適用されると最後の引数を関数の戻り値として返します。

上記コードをラムダ式にすると以下の様になります。

```
f := λ xyz.z
 λ xy.fxyy
```

2 回目の関数定義においてクロージングされるスタックに 1 回目の関数定義が含まれることに

注意してください。
そのため 2 回目に定義される関数の本体で 1 回目の関数を呼び出すことが出来ます。

Grass 全体の構文を BNF で表すと以下のようになります。

```
〈program〉 ::= 〈def〉 | 〈program〉 "v" 〈def〉 | 〈program〉 "v" 〈app〉
```

Grass 抽象機械によって初期化されるスタックの内容を初期環境（プリミティブ）と言いますが、初期環境はスタックの上から順に以下の様になります。

Out	引数を1つ受け取り文字として標準出力に出力する関数。引数をそのまま返す。
Succ	次の文字を返す。ただし256との剰余。ラムダ計算のsuccとは別物なので注意。
"w"	文字としての"w"。文字コード119で表される。関数としても呼び出せる（後述）。
In	標準入力から1文字入力する。EOFの場合は引数をそのまま返す。

文字は関数として呼び出すことも出来て、その場合引数を 1 つとり引数と文字が等しければ TRUE を、等しくなければ FALSE を返します。これらはラムダ計算と同じ定義なので、TRUE と FALSE は以下の様になります。

```
TRUE  := λ xy.x
FALSE := λ xy.y
```

Grass 抽象機械がすべてのコードをパースし終わると、スタックの 1 番目にある関数にその関数自体を適用し、関数の実行が終了すると Grass プログラムも終了します。この自分自身の適用は Grass 抽象機械が自動で行うのでコード中に書いてしまってはいけません。
また、この性質を上手く使うと Y コンビネータ*1 を使わなくても再帰を作ることが出来ます。

正直堅苦しい定義ばっかで疲れたから何かGrassのコードくれ

一般に新しい言語を習得する時は、"Hello,world"から始まることが多いのではないでしょうか。しかし esolang は大抵の場合において"Hello,world"ですら非常に可読性の低いコードが完成します。これでは言語の習得どころではありません。
なので本記事では本当に短いコードから紹介していきます。

*1 残念ながら筆者はまだ Y コンビネータについて理解しきれていません。なので本記事では Y コンビネータを用いた再帰については触れません。あしからず。

まず初めに w とだけ出力するコードです。

```
wWWwwww
```

関数定義が 1 つあるだけです。しかも関数の本体には関数適用が 1 回しかありません。つまり、エラーの発生しない最小の Grass コードです。

この関数が最終的に実行されるわけですが、その時にはスタックの 2 番目には初期環境での Out があり、4 番目には"w"があります。つまり関数 Out に文字 w を適用するわけです。Out の戻り値は引数と等しいので、表示した後スタックには w が積まれて終了します。

前のコードだとあまりに味気なかったので、次は再帰を用いて w を無限に表示するコードです。

```
wWWwwwwWWww
```

上の関数をラムダっぽく書くとこうなります。

```
F := λ f.FALSE(Out "w")(ff)
```

この関数が定義され（スタックにプッシュ）実行されると、先ほどの関数にそれ自身が適用されるので、以下のように再帰的に展開、実行されます。

```
FF → (λ f.FALSE(Out "w")(ff))(λ f.FALSE(Out "w")(ff))
   → FALSE(Out "w")((λ f.FALSE(Out "w")(ff))(λ f.FALSE(Out "w")(ff)))
   → FALSE(Out  "w")(FALSE(Out  "w")((λ f.FALSE(Out  "w")(ff))(λ f.FALSE(Out
"w")ff))))
   → ...
```

ただし関数適用する毎にスタックに引数がプッシュされるので、実際の計算機で実行すると有限回でスタックオーバーフローを起こします。

あまり解説を書く事もないので、様々なコードを列挙します。どのように動作しているのか考えてみてください。ただ、あまり最適化はされていません（特に最後のコードはかなり苦し紛れで書きました）。

見やすいように関数定義ごとに改行を入れてあります。（言語仕様の特性上、結局見づらくなりますが。）

x と表示する

```
wWWWwwwwWWWw
```

1+1 の答えを w の数で表す

```
wwWWwv
wwwwWWWwwWwwWWWWWWwwwwWwwv
wWWwwwWwwwwWwwwwwwWwwwwwwwww
```

2*2 の答えを w の数で表す

```
wwWWwWWWwv
wwwwWWWwwWWWWWwWwwwv
wWWwwwWwwwwWwwwwwwWwwwwwwwww
```

wxyz と表示する

```
wv
wWWWWwwwwwWWWWWwWWWWWWwWWWWWWWv
wwWWwwWWWWWWwwWWWWWWWWWwWWWWWWwWwwWwwwwwwwwWwwww
wwWwwwwwv
wWWwwWwwwwwwww
```

Hello, world と表示する[*1]

```
wwv
wwwWWWwwWwwWWWWwv
wWWwwwWwwv
wWwwwWwwv
wWWwWWWWWwv
wWWWwwwwWWWWwWWWWwWWWWWWwWWWWWWWWwWWWWWWWWwWWWW
WWWWw
Wwwwwwwwwv
wWWWwwwwwWWWWWWwWWWWWwWWWWWWWWwWWWWWWWWwWWWWWWW
```

*1http://vipprog.net/wiki/プログラミング言語/Grass.html より引用。

WWwWwwwwwwv
wWWWWwwwwwwWWWWWWwWWWWWWwWWWWWWWWwWWWWWWWWWwWW
WWWWWWWWwWwwwwwwwv
wWWWWWwwwwwwwWWWWWWWWwWWWWWWWWWwWWWWWWWWwWWWWWW
WWWWwWWWWWWWWWWwWwwwwwwwv
wWWWWWWWwwwwwwwwWWWWWWWWwWWWWWWWWWwWWWWWWWWw
WWWWWWWWWWWwWwwwwwwv
wWWWWWWWWwwwwwwwwwWWWWWWWWwWWWWWWWWWwWWWWWWWW
WWwWWWWWWWWWWWwWWWWWWWWWWWWWwWwwwwwwwv
wWWWWWWWWwwwwwwwwwwWWWWWWWWWwWWWWWWWWWwWWWWWW
WWWWWwWWWWWWWWWWWWWwWwwwwwwv
wWWWWWWWWWWwwwwwwwwwwwWWWWWWWWWWwWWWWWWWWWWwW
WWWWWWWWWWWWWwWwwwwwv
wWWWWWWWWWWWwwwwwwwwwwwwwwwwwWWWWWWWWWWWWwWWWWWWWWW
WWWwWWWWWWWWWWWWWWWwWWWWWWWWWWWWWWwWWWWWWWW
WWWWWWWwWwwwwwwwv
wWWWWWWWWWWWwwwwwwwwwwwwwwwwwWwwwwwwwwwwwwwwwwwwwWWWw
wwwwwwwwwwwwwwwwwwWwwWWWWWWWWWWWWWWWWWWWwv
wWWwwwwWWWwwwwwwwwwWWWWwwwwwwwwwWWWWWwwwwwwwwwwwww
WWWWWWWwwwwwwwWWWWWWWWwwwwwwwwwwwwWWWWWWWWwwwwwww
wwwwwwwwwwwwwwwv
wWWWWWWWWWWWWWWWWWwwwwwwWwwv
wWWWWwwwwwwWWWWWwwwWWWWWWwwwwwwwWWWWWWWwwwwwwwwW
WWWWWWWWwwwwwwwwwwwwwwwWWWWWWWWWWwwwwwwwwwwwwwv
wWWwWWWWWWWWwWWWWWWwWWWWWWWwWWWWWWWWwwwwv
wWWWwWWWWWWWWWwWWWWWWWWWWwWWWWWWWWWwv
wWWWWWWWWWWWWWwv
wWWWWwwwwwwwwwwwwwwv
wWWwWWWWWWWwWWWWWWwWWWWWWWwWWWWWWWWwwwwv
wWWWwWWWWWWWWWwWWWWWWWWWWwWWWWWWWWWwv
wWWWWWWWWWWWWWwv

A happy new year!! と表示する

wwWWwWWWwv
wwWWWwwWwwWWwv
wWWWwwwwwWwwv
wWWwWWWWWWWwv
wWWWWwwwwwwWwwv
wWWWWWWwwwWwwv
wWWWWWWwwwWWWWWWWWwWwwwv
wWWWWWWWwwWwwWWWWwWWWWWWwWWWWWWWWWwv
wWWWWWWWWWWWWwv
wWWWWWWWWWWWWWwv
wWWWWwwwwwwwwwwwwwwv
wWWwWWWWWWWwWWWWWWwWWWWWWwWWWWWWWwwwwv
wWWWwWWWWWWWWWwWWWWWWWWWWwWWWWWWWWWwv
wWWWWWWWWWWWWWwv

```
wWWWWWWWWWWv
wWWWWWWWv
wWWWWWwWWWWwWWWWWWWWWWWWwWWWWWWWwWWWWWWwWWWWW
WWwwwwwWWWWWWWWWwwWWWWWWWWWwWWWWWWWWWWwWWWWW
WWWWWWwWWWWWWWWWWWWWWwwwwwwwwwwwwwwwwwwwwwwww
wWWWWWWWWWWWWWWwWWWWWWWWWWWWWWWWWWWwWWWWWWW
WWWWWWWv
wWWWWWWwWWWWWwWWWWWWwWWWWWWWWwWWWWWWWWWWWWWw
WWWWWWWWWwWWWWWWWWWwwwwwWWWWWWWWWWwwwwwwwwwwwww
wwwwwwwwwwwwwWWWWWWWWWWWWWWWWWwWWWWWWWWWWWv
wv
wWWwwwwwwwwwwwwwwwwwwwwwv
wWWWWWWWWWwWWWwWWWWWWWWWWwWWWWWWWWwWWWWWWW
WWWwwwwWWWWWWWWWWWWwWWWWWWWWWWWWwwwwwwWWWWWWW
WWWWWWWWWWWWwwwWWWWWWWWWWWWWWWwWWWWWWWWWWWW
WWWwWWWWWWWWWWWWWWWWWwWWWWWWWWWWWWWWWWv
wWWWWWWWWWWWWWwWWWWWWWWWWWWWWWWwWWWWWWWWWWW
WWwWWWWWWWWWWWWWWWWv
wWWWWWWWWWWWWWwWWWWWWWWwWWWWWWWWwWWWWWWwWWWW
Ww
```

おわりに

今回はマンデルブロ集合ではなく Grass という関数型言語について書きました。正直筆者も w と W と v がこうも連続すると区別がつかず、数を数えるので精一杯でした。

しかし、紙とペンを持ってラムダ式を延々と書きつづければそのうち動くようになります。筆者も 2 時間半格闘して最後に紹介した A happy new year!!と表示するプログラムを書き上げました。人間成せば成るものです。

ところで Grass 抽象機械なんてねぇぞゴルァ！って方が大半だと思いますが、単純なインタプリタなら比較的楽に記述することが出来ます。

筆者は Common Lisp[*1] を勉強する為に Grass のインタプリタを Common Lisp で書きました[*2]。インタプリタ自体のコード量は 110 行程度[*3] でした。インタプリタを書いているときは Lisp の（）と Grass の wWv が大量にあり、疲れました。

とはいえそんなこと面倒だと思う人は多いと思います。Grass の公式ページ[*4] に先人が作ったインタプリタがあります。

---

*1 関数型言語である Lisp の方言。ANSI によって標準化された。

*2http://www.coins.tsukuba.ac.jp/~s0911476/gri.cl にあります。ライセンスは 3 または 4 条項 BSD ライセンスなのでご自由にどうぞ。現時点で動作する実装は GNU CLISP のみです。

*3 現在は 450 行前後まで膨れあがっている。

*4http://www.blue.sky.or.jp/grass

数種類の言語で実装されているので、自分にあったものを選ぶといいと思います。

もし Grass 興味が出てきたなら独自拡張した Grass を作ってみても面白いかもしれません。ちなみに筆者は密かにマルチスレッド化を目指しています。

Grass でなくても、他の esolang をやってみるのもいいかもしれません。

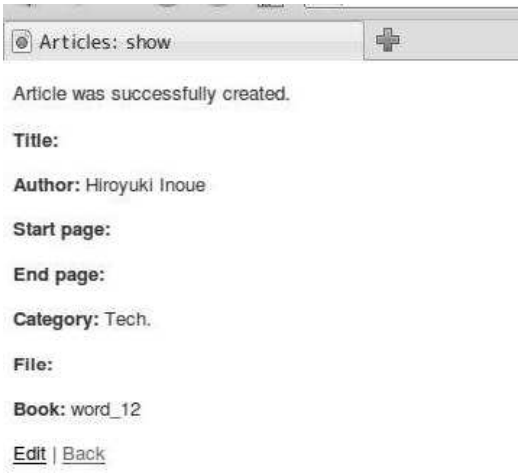
入門 Ruby on Rails Vol.5

文 編集部 いのひろ

新年あけましておめでとうございます。今年もよろしくお願いいたします。

Validation の設定

今回はフォームに Validation（値チェック）をつけてみます。例えば、現在のコードのままだと、記事新規作成（articles/new）のとき、タイトル（Title）が空欄でも作成する事ができてしまいます。



また、開始ページ（Start page）、終了ページ（End page）では、私たちは数字（Integer 型）の入力を期待していますが，文字も入力できちゃいます（文字が入力された場合は「0」に変換されていますが）。

このような意図しない入力に対しての Validation（値チェック）をつけてみようとおもいます。

Validation をつけるのは簡単です。Model に以下のように書きます。

```
class Article < ActiveRecord::Base
  validates_presence_of :title
  validates_numericality_of :start_page, :end_page
  ...
```

app/models/article.rb

これで、:title には「テキストボックスが空欄でないかのチェック」、:start_page と:end_page には

「入力されたデータが数字かどうかのチェック」がされるようになりました。

本来はこれだけで問題ないのですが、View で「collection_select （ドロップダウンリストを生成するメソッド)」を利用している関係で、articles_controller.rb、L61 付近（@article.save を評価する if 文の else の下）に以下の三行を書く必要があります。

```
if @article.save
  ...
else
  @author = Author.find( :all )
  @category = Category.find( :all )
  @book = Book.find(:all)
  ...
```

app/controllers/articles_controller.rb

model に書いた Validation（値チェック）が実行されるのは、@article.save が実行されたときで、値チェックでエラーが発生すると、save アクションは false を返します。そのため else の中に飛んでくるわけですが、view の@author, @category, @book は articles_controller.rb の new アクション（L 付近）で定義されたインスタンス変数なので、create メソッドの中で再定義する必要があります。その為の三行です。

これらを記述したら実際に挙動を確かめてみましょう。

```
$ script/server
```

http://localhostt:3000/articles/new にアクセス。テキストボックスに何も入力しないで「create」をクリックしてみます。右の画像のように値チェックがきちんと動作したでしょうか？

エラーメッセージなどを日本語化するには多少手間がかかるため、今回は省略させていただきます。

:title のテキストボックスに適当に文字を入力すれば、値チェックが「文字が入力されている」と判断して、エラー表示が取り消されます[*1]。

但しこの状態だと:start_page や:end_page にマイナスの値等が入力されても、そのまま登録できてしまいます。また:start_page よりも:end_page の方が小さい、という訳の分からない情報も保存できてしまいます。さらに値チェックを強化しないといけません。Rails が値チェックのときに呼んでいるメソッドにエラーの条件を付け足します。article モデルを開いて、validate メソッドを定義し、中に次の二行を書きましょう。

```
private
def validate
  errors.add( :start_page, "は 1 以上である必要があります。" ) if start_page.nil? || start_page <= 0
  errors.add( :end_page, "は Start Page と同じか、それより大きい必要があります" ) if end_page.nil? || end_page < start_page
end
```

app/models/article.rb

一行目は、Start Page（:start_page）が 1 以上でないと発生するエラーです。二行目は、End Page（:end_page）が Start Page（:start_page）と同じかそれ以上出ないと発生するエラーです。これが

*1Ajax とかで出来たらさらにカッコいいですね

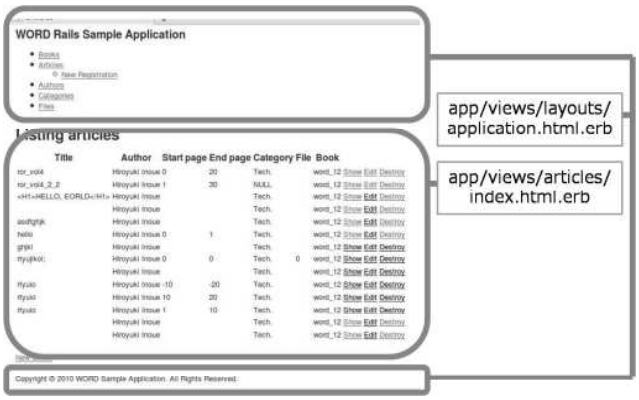
うまく動いているか、確認してみましょう。

まとめの方向へ

連載も五回目となりましたが、結局中途半端な感じになってしまいました。WORD のアーカイブアプリという名目でしたが、実際に PDF を保存する機能も実装できませんでした。さらに現状だと Articles や Books などのページがバラバラな状態なので、見た目というか、ページ間の連結くらいは作っておきたいところです。

まずはそれぞれの View のベースとなる HTML ファイルを作りましょう。さらにアプリケーションのトップディレクトリにアクセスしたときに、Articles や Books に簡単にアクセスできるようにしてみます。

app/views/layouts/application.html.erb というファイルを作ってください。既にこのディレクトリ（app/views/layouts）には、articles.html.erb や books.html.erb などのファイルがあります。これらのファイルは、app/views/articles/index.html.erb や new.html.erb などのページが埋め込まれるベースとなる html.erb ファイルになっています。



article.html.erb や books.html.erb で同じレイアウトを用いる場合などに、同じレイアウトを複数の html.erb にコピーして書くと、レイアウト部分を修正したときの手間が大変なものになります。このようなときに、articles.html.erb や books.html.erb の代わりとして、application.html.erb を導入することで、サイト内で一貫したレイアウトを共有する事ができます。

```
<html>
  <head>
    <title><%= params[:controller].to_s %></title>
    <%= stylesheet_link_tag 'scaffold' %>
  </head>
  <body>
    <h2>WORD Rails Sample Application</h2>
    <ul>
      <li><%= link_to "Books", :controller => "books", :action => "index" %></li>
      <li>
        <%= link_to "Articles", :controller => "articles", :action => "index" %>
        <ul>
          <li><%= link_to "New Registration", :controller => "articles", :action => "new" %></li>
        </ul>
      </li>
      <li><%= link_to "Authors", :controller => "authors", :action => "index" %></li>
      <li><%= link_to "Categories", :controller => "categories", :action => "index" %></li>
      <li><%= link_to "Files", :controller => "data_files", :action => "index" %></li>
    </ul>
    <hr />

    <%= yield %>

    <hr />
    <p>Copyright &copy; 2010 WORD Sample Application. All Rights Reserved.</p>
  </body>
</html>
```

app/views/layouts/application.html.erb

application.html.erb を作成して、上の HTML を書きます。title タグの中には、呼び出し元の controller 名を挿入するコードが書いてあります。さらに、body の中で各ページへのリンクを作っています。

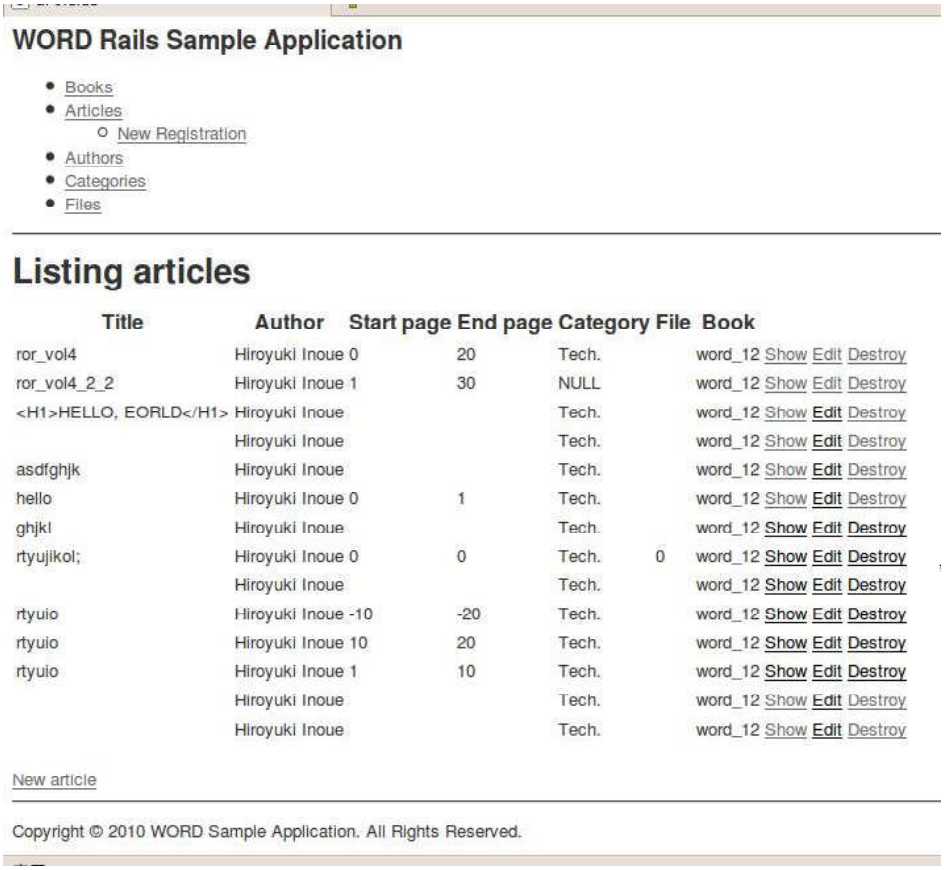
```
<%= link_to "Hello", :controller => "world", :action => "hogehoge" %>
```

というコードは、

```
<a href="アプリケーションルート/world/hogehoge">Hello</a>
```

という HTML を出力します。パスの計算は Rails がやってくれるので、たとえば「アプリケーションルート/」にいても、「アプリケーションルート/world/hogehoge」にいても、「<%= link_to "hello", :controller => "world", :action => "hogehoge" %>」から生成された a タグは、正しく「アプリケーションルート/world/hogehoge」にリンクされます。

その下の「<%= yield %>」は、app/viesw/なにか/どれか.html.erb が埋め込まれる事を示しています。先ほどの図で示したように、controller がレンダリングする view を選択し、それを app/views/layouts/application.html.erb に埋め込んでブラウザに返します[*2]。



*2 判りやすいように application.html.erb の「<%= yield %>」の前後に「<hr />」タグを挿入しておきました

既に app/views/layouts に存在する他の html.erb ファイルを削除するか、リネームしてしまいましょう。サーバーを実行して表示を確認してみます（/articles にアクセス）。

若干レイアウトがおかしい気がしなくもないですが、そこはご愛嬌で。これで各ページへの行き来が簡単になったと思います。

さらに、「アプリケーションルート/」にアクセスしたときの挙動を変更してみましょう。 デフォルトのままだと Ruby on Rails が正常に動いている事を示すページが表示されるはずです。top コントローラを導入し、アクセスされたときのルーティングを変更してみます。

```
$ script/generate controller top
      exists  app/controllers/
      exists  app/helpers/
      create  app/views/top
      exists  test/functional/
      create  app/controllers/top_controller.rb
      create  test/functional/top_controller_test.rb
      create  app/helpers/top_helper.rb
```

top_controller には index アクションを定義しておきます。が、中では何もしません。

```
class TopController < ApplicationController

  def index
  end

end
```

app/controller/top_controller.rb

top コントローラの index にアクセスされたときに呼ばれる app/views/top/index.html.erb を作成し、何か適当に書いておきます。これがトップページとなるページです。

```
<h1>Welcome to WORD Sample Application</h1>
```

app/views/top/index.html.erb

これで「アプリケーションルート/top」もしくは「アプリケーションルート/top/index」にアクセ

スすると、「Welcome to WORD Sample Application」と表示されるはずです[*3]。

最後にルーティングの設定をします。今回は「/」にアクセスされたときに、そのリクエストを「top/index」へのリクエストとして処理する様にルーティングを変更します。config/routs.rb の下の方に一行書き足します。

```
  map.connect '', :controller => 'top', :action => 'index'

  # Install the default routes as the lowest priority.
  map.connect ':controller/:action/:id'
  map.connect ':controller/:action/:id.:format'
end
```

config/routs.rb

「map.connect '', :controller ~」は、「/」にアクセスされたときに、「top/index」に飛ばすという処理です。これで（テスト時であれば）「http://localhost:3000/」にアクセスしても「Welcome to WORD Sample Application」と表示されるはずです。

*3 先ほど確認したのと同じように、上下に app/views/layouts/appliation.html.erb の html も表示されているはずです

まとめ

最後はかなり駆け足で、解説が足りないのではないかと思います。ぐだぐだですみません。

Ruby on Rails による Web アプリケーション開発を少しでも知ってもらえたら良いと思い、書いてきました。Rails は Web アプリケーション開発者のかゆいところに手が届く機能がたくさんあり、開発していて楽しいです。皆さんも是非この楽しさを感じてもらえれば良いなと思います。

この連載をきっかけに Ruby on Rails に少しでも興味をもっていただければ幸いです。

参考文献

Rails を勉強するには以下の書籍がおすすめです。

Rails によるアジャイル Web アプリケーション開発（単行本）

Sam Ruby（著）, David Heinemeier Hansson（著）, Dave Thomas（著）, 前田 修吾（翻訳）

http://www.amazon.co.jp/dp/4274067858/

入門 zsh

文 一七夜月

皆さんこんにちは、お久しぶり、初めまして。一七夜月です。突然ですが皆さんはどんなシェルを使っていますか？ bash でしょうか、それとも tcsh でしょうか。コマンドプロンプトや PowerShell なんて人もいるかもしれません。シェルもそれぞれ派閥があり、Vim と Emacs のように宗教戦争が繰り広げられているわけですが、今回はシェル戦争に新たなる火種を投げ込んでみたいと思います(ｳｿ)。

皆さんは zsh というシェルをご存じでしょうか。zsh というのは、その名の通り[*1] 最強のシェルを目指して作られたシェルです。後発であるメリットを生かしてほかのシェルのいいとこ取りをしたあげく、ほかのシェルにない機能を多数備えています。今回はその zsh の機能を少しずつ紹介していきたいと思います。

zsh は Coins の iMac にも[*2] インストールされているので、コマンドラインで"zsh"と入力するこですぐにでも利用できます。どうしてもログインシェルを zsh にしたいという人は Coins-admin の人に頼むと変更してもらうこともできます。間違っても.bashrc に"zsh"と記述しないように[*3] お願いします。

まず最初に

Bash に.bashrc があるように zsh にも.zshrc があります。もちろん内容は好きなようにかけますが、とりあえず最低限以下を記述しておきましょう。内容は追って説明していきます。

```
# ~/.zshrc

#コマンド履歴を残す設定
HISTFILE=~/.zsh_history
HISTSIZE=10000
SAVEHIST=10000
setopt hist_ignore_dups   # コマンドが重複して記録されるのを防ぐ
setopt share_history      # 履歴を共有する

# プロンプトの表示設定
PROMPT="%n@%m%% "
RPROMPT="[%~]"
```

*1 "z"はアルファベット最後の文字のため最強を意味したい場合使われることが多々あります。
*2 残念ながら orchid-calc の Linux マシンには執筆段階では入っていません。
*3 .bashrc は bash が起動する度に呼び出されるため、内容によっては無限ループに陥る可能性があります。.bash_profile ならばログイン毎に呼び出されるためそのような心配はありません。

```
SPROMPT="%r is correct? [n,y,a,e]: "

autoload -U compinit   # コマンド補完を有効化
compinit
setopt auto_cd         # ディレクトリ名で移動
setopt auto_pushd      # 移動履歴を利用する
setopt correct         # コマンドのタイポを推測する
setopt nolistbeep      # 補完時にビープ音を鳴らさない
```

最初は見た目から

何事も格好をつけなければ始まりません。とりあえずプロンプトを見やすくしてみましょう。上記の.zshrc の

```
PROMPT="%n@%m%% "
RPROMPT="[%~]"
SPROMPT="%r is correct? [n,y,a,e]: "
```

の部分がそれに当たります。なぜ 3 つもあるかというと、左側のプロンプト、右側のプロンプト、それに、コマンドを間違った際に表示されるプロンプトの設定というわけです。

左側のプロンプトというのは、皆さんがよく目にしている

```
kano:~> _
```

というのがそれに当たります。ただユーザー名だけでは味気ないのでここでは

```
kano@Kudryavka% _
```

のようにマシン名を表示するようにしました。

そしてここからが zsh の本領発揮、右プロンプトです。これはその名の通り右端にくっついているプロンプトのことですが、これの便利なところは、コマンドが長くなって干渉するようになると、自動的に消えてくれることです。それが何の役に立つのかというと、皆さんにもこんな経験はありませんか？

```
kano:/usr/lib/python2.5/site-packages/apt/progress> _
```

このようにプロンプトにディレクトリ名を表示させていると、深い階層に入れば入るほどプロンプトが長くなってしまいます。しかし表示させないと今どこにいるのかを知るためにいちいちコマンドをたたかなければいけなくて不便ですね。

ここで zsh の右プロンプトにパスを表示させておけば、

```
kano@Kudryavka% _                    [/usr/lib/python2.5/site-packages/apt/progress]
```

これですっきりとしました。しかも長いコマンドを入力する場合は自然と消えてくれるため邪魔になりません。

コマンドを間違った際のプロンプトというのは、これは.zshrc に記述した

```
setopt correct
```

の設定と同時に動作するのですが、zsh にはコマンドを間違った際に正しいコマンドを推測し、補完してくれる機能があります。その際に出現するプロンプトのことです。

```
kano@Kudryavka% sud aptitude search zsh                                [~]
sudo is correct? [n,y,a,e]: y
i   zsh                          - 多機能シエル
p   fatrat-czshare               - fatrat plugin allowing download and upload to
・・・
```

そんなコマンド間違えるなという心ない突っ込みは無しですよ^^　急いでいるときに限って「ls」を「sl」とタイポしてしまうものです。

履歴の有効活用

zsh の特徴の一つに強力な履歴機能があります。履歴機能自体は bash や tcsh にもありますが、zsh の履歴はひと味違います。

```
HISTFILE=~/.zsh_history
HISTSIZE=10000
SAVEHIST=10000
setopt hist_ignore_dups
setopt share_history
```

上が先ほど.zshrc に記述した履歴にに関する部分です。最初の 3 行で履歴を 10000 件 home 以下の.zsh_history というファイルに記録するように指定しています。

4 行目は同じコマンドが何度も叩かれたときに重複して記録するのを防ぐための設定です。これを設定しておけば sl コマンド[*4] で何回遊んでいても履歴には一回分しか残らないので安心です。ただし連続して叩いた分しか纏められないので注意してくださいね。

最後の行はこれも zsh を最強たらしめている機能を有効にする設定、コマンドの履歴共有です。

screen[*5] を使っていてもいなくても、シェルを複数立ち上げることはよくあると思います。その複数のシェルでコマンド履歴を共有できたら便利だと思いませんか？　zsh ならそれが可能です。

---

*4 端末に SL を走らせるコマンド。Mac だと GUI 上を走ります。

*5 端末エミュレーション機能を持つ画面管理ソフトウエア。

なんでも補完

zsh の数ある機能の中でも一番かゆいところに手が届く機能というのはこのなんでも補完してくれる機能ではないでしょうか。何でもといっても抽象的すぎるので具体的にパス補完、ワイルドカード、オプション補完の 3 つについて説明します。本来は全く関係ない別々の機能なのですが、ユーザーがいちいち長い入力をしなくても補完してくれるという意味合いでひとまとめにして説明します。

.zshrc のこの部分が便利な補完を有効にする設定です。

```
autoload -U compinit
compinit
```

まずはパス補完について。パス補完といっても/etc/apa ［tab］と入力して/etc/apache2 と補完してくれる機能ではありません。その程度の機能ならどのシェルにもありますが zsh のパス補完はひと味違います。たとえば apache2 の再起動がしたいとき、どうしますか？

```
kano:~> sudo /etc/init.d/apache2 restart
```

とコマンドを打つ必要があります。ところが、ここでたとえば tcsh のパス補完を使おうと思った場合、/etc 以下には init.d と initramfs-tools という 2 つのディレクトリが存在するため一意にしようとすると/etc/init.までは手で入力しなければなりません。ところが zsh では次のように変態的な補完ができます。

```
kano@Kudryavka% sudo /e/i/a [tab]                                    [~]
kano@Kudryavka% sudo /etc/init.d/apache2 _                           [~]
```

どうでしょうか。zsh ではこのようにディレクトリ名の最初の一文字ずつを入力しただけで補完してくれます。どれだけ長いディレクトリ名が連続していたとしても最初の一文字を入力するだけで後は zsh 任せにできます。もちろん一文字でなくともかまいません。

上記の補完によって楽になるコマンド入力をさらに加速させる zsh の機能、それがワイルドカードです。

たとえば、home 以下のどこかにおいたまま所在がわからなくなっているファイル"Sample.txt"を Vim で編集したいとします。この場合普通なら home 以下を find にかけて見つけたファイルを～とするところですが、zsh なら一発です。

```
kano@Kudryavka% vim ./**/Sample.txt [tab]                            [~]
kano@Kudryavka% vim ./documents/2008/12/Sample.txt                   [~]
```

この**はとても便利で、find コマンドを使う機会はなくなってしまうかもしれません。

便利な補完機能の 3 つめ、オプション補完は、その名の通りコマンドの引数をオプション込みで補完してくれます。例えば、tar コマンドで Sample.tar.bz2 を展開したいとき

```
kano@Kudryavka% tar -[tab]                                          [~]
A  -- append to an archive
c  -- create a new archive
f  -- specify archive file or device
t  -- list archive contents
u  -- update archive
v  -- verbose output
x  -- extract files from an archive
```

といった具合にオプションを補完、候補を説明付きで表示してくれます。

この補完はコマンドに対応する設定ファイルが存在し、自分で作ったコマンドに対しても設定ファイルを書くことで補完を適応させることができます。

ディレクトリあれこれ

ディレクトリを移動する際、"cd　"の 3 文字を入力するのも煩わしい場合は auto_cd を設定しておきましょう。そうすればディレクトリ名を叩くだけで移動できるようになります。ただし、ディレクトリ名と同じコマンドが存在する場合はそちらが優先されるのできちんと"cd　"を叩きましょう。

zsh のディレクトリに対する便利な機能、それは移動履歴です。深い階層のディレクトリを行ったり来たりすることはよくあることですが、そのたびに長いパスを入力するのは面倒ですし、履歴を使って探すのも冗長です。そんなときに活用するのがこの機能。使い方は至って簡単。
"cd - [tab]"と入力するだけです。

```
kano@Kudryavka% cd -[tab]                                           [~]
0 -- /home/kano
1 -- /home/kano/documents
2 -- /var/log
3 -- /usr/lib/python2.5/site-packages
4 -- /etc/apache2/sites-available
5 -- /usr/share/zsh/functions/Completion
```

候補を表示させたら、後はその番号を入力するだけでそのディレクトリに移動できます。どうですか、地味に便利でしょう。相対パスで移動したディレクトリも絶対パスで記録されるのでどこからでも目的のディレクトリに飛べます。

便利な小技

・**Meta（Alt）-h**

現在入力中のコマンドを man 参照できます。man から戻ってくると入力中の内容が復活します。

・**Meta（Alt）-?**

現在入力中のコマンドを which 参照できます。which から戻ってくると入力中の内容が復活します。

・**Meta（Alt）-q**

現在入力中のコマンドをバッファにスタックします。あるコマンドを入力中に別のコマンドを入力したい場合に使います。コマンドが終了するとスタックされたコマンドが復活します。コマンドのオプションを調べたり、引数に渡すファイルを調べたりするのに使えますが、簡単なものならオプション補完と Meta-h、ワイルドカードですんでしまうため便利なのにあまり活躍の場がない小技かもしれません。

おわりに

どうでしょうか。少しでも zsh を使ってみようという気になってもらえたでしょうか。zsh にはほかにも複数行コマンドが簡単に編集できたり、キーバインドを Vim 風 Emacs 風に切り替えられたり、zsh の機能のみで書かれたエディタ"zed"なんてものもあったりと、膨大な機能を有します。そのため、zsh のマニュアルの最初の一行が「zsh　は多くの機能を持っているので、マニュアルはいくつかのセクションに分かれています。」と始まっているほどです。

ですがそんな機能をすべてマスターしなくとも zsh は十分に便利なシェルです。この機会に皆さんが zsh を使うようになってくれたら幸いです。

参考文献

http://www.zsh.org/　zsh オフィシャルサイト
http://journal.mycom.co.jp/column/zsh/index.html　マイコミジャーナル　漢の zsh

TOEIC の点の取り方

文 編集部 中退アフロ佐々木

みなさまこんにちわ。情報科学類 2 年の佐々木和平と申します。早稲田辞めてきました。それと、このたび WORD に入れてもらえることになりました。でもここは何をする団体なのかまだよく分かっていません。こんな僕でも何か書いて良いと言われたので、名刺代わりに僕の好きな TOEIC（トーイック）のことについて書かせて頂きたいと思います。役に立つかは分かりませんが、ご笑読いただければ幸いです。

■ TOEIC ってなに

TOEIC の概要についてザックリと説明します。TOEIC はアメリカの教育機関 ETS が開発した、英語能力を測る試験です。ETS というのは、日本でいう大学入試センターの大規模版と考えてください。TOEFL や SAT なども ETS が一手に作問しています。そういうわけで、表向き TOEIC は世界基準の英語テスト……ということになっています。ですが、実際は日本と韓国だけで受験者の 9 割を占めているという現実があったりします。

TOEIC はリスニングセクションとリーディングセクションから、それぞれ 100 問ずつ出題されます。TOEFL-iBT のように、ライティング・スピーキングの試験はありません。全てマークシート上で解答します。合格・不合格ということは無く、10 ～ 990 点のスコアで評価が下されます。一回の試験時間は 2 時間なので、集中力を切らさないことが大事です。最後の方は飽きて嫌になってくるので、集中力は侮れないポイントです。

ところで、TOEIC 好きを名乗るくせにこういうことを言うのもナンですが、個人的には TOEIC は欠陥が多いテストだと考えています。その理由は色々あるのですが、端的に言えば現実の英語力を必ずしも正確に表していないということです。つまり、TOEIC 専門のトレーニングを積んでしまえば、実際の英語力が大した事なくても、ある程度高いスコアが叩きだせてしまうのです。「英語が出来る⇒ TOEIC の点が高い」は真ですが、その逆「TOEIC の点が高い⇒英語が出来る」というのは必ずしも成り立たないと、僕はそう思っています。それではどうして、欠陥が多いと思っていながらに TOEIC の受験を皆様にお勧めするのでしょうか？その理由については、次でご説明させていただきます。

■ TOEIC の美味しい点

筑波大学理系の皆様にとっての最大のメリットは、ズバリ大学院入試に使えるということでしょう。システム情報工学研究科では、平成 23 年度入試から英語の試験を TOEIC もしくは TOEFL のスコア提出に一本化することが決まりました。具体的に何点とれば良いのかという基準は公表されていませんが、平成 22 年度入試の場合は次のようになっています。

TOEFL-iBT	TOEFL-PBT	TOEFL-CBT	**TOEIC**
79	550	213	**730**

この得点に達していれば、「英語」の得点が満点換算になります。（平成 22 年度募集要項 25 ペ

ージ)。どんな試験でもそうですが、「満点」というのは滅多に取れるものではありません。英語の筆記試験を受けるとなれば、どんなに英語が得意な人でも、一個や二個のミスはするものです。それを TOEIC のスコアで代用できるというのは、非常に有り難い措置だと言えましょう。前述の通り、平成 23 年度の入試で何点取れば良いのか（上記のような基準点が存在するのか）は今の所わかりませんが、大学側が TOEIC730 点を 1 つの基準として考えていることは明らかだと思われます。ですので、院進学を検討なさっている皆様におかれましては、まずは 730 点に向けて勉強して頂きたいと思うのです。ちなみに、「平成○○年○月以降に受験したもの～」という条件（通常は 2 年以内）もあるはずなので、過去に受けた事があったとしても、再度確認する事をおすすめします。

もちろん下級生にもメリットはあります。TOEIC730 点以上（情報科学類の場合）のスコアを提出することで、必修科目の英語 II の単位が認定してもらえます。私は既に英語 II の単位を取ってしまったので利用しませんでしたが、1 年生や再履修生にとっては嬉しいのではないかと思います。その辺のことが知りたかったら、履修要覧を参照してください。

それでは、大学院に行かず、英語 II の単位も取り終えた人には何のメリットも無いのでしょうか？いえいえ、決してそんなことはありません。TOEIC で高得点を持っていれば、大事な大事な就職活動に大きく役立つことになるのです。就職活動では人柄が重視されるのは間違い無いことですが、それ以前に語学力は必要不可欠な能力だと言われています。TOEIC のハイスコアを持つと社会でどのように評価されるかの例として、先ほどの表に近いのですが次のような換算表を示します。

TOEFL-iBT	TOEFL-PBT	TOEFL-CBT	英検	TOEIC
79	550	213	準一級	730

これらは企業では等価のものとして扱われています。その辺のことが知りたかったら、グーグルで「TOEIC　換算」とか入れてみてください。では、あえてこの中から TOEIC を選ぶ理由はあるのでしょうか？それはズバリ、簡単だからです！冒頭で述べた事をぜひ思い出してください。TOEIC は英語力が無くても点が取れてしまう欠陥商品です。その欠陥を逆手に取って勉強すれば、英語を武器に就職活動に立ち向かえるのです。個人的な話ですが、私は TOEIC825 点と英検準一級を持っていますが、TOEFL-iBT は 72 点しか取れませんでした（換算表では 79 点以上取れてもおかしくないハズ……）。自分なりに試行錯誤した実感として、TOEIC が一番簡単だと自信を持って言えます。また、英語の指標として企業が最も重視するのも TOEIC であると言われています。つまり、TOEIC は簡単なのに評価が高いという、三段論法的に美味しい試験なのです。

ここまで述べたように、TOEIC は費用対効果の高い試験である事は分かって頂けたと思います。それでは、どういう方法で TOEIC の点が取れるのかを次でご説明しましょう。

■勉強のやり方

（STEP1：計画を立てよう）

どんな目標を立てるにしても、計画は大事です。大学受験だろうが TOEIC だろうが、それは変わりません。いつまでに何点必要なのかを計算してみましょう。WORD を読んでいる人の殆どが情報の学生であるという事実を鑑みて、とりあえず大学院入試へのモデルプランを立ててみたいと思います（情報の人は 8 割くらい院に行くと思うので、需要が一番ありそうなプランです)。

TOEIC の点の取り方

あなたは平成 22 年 1 月現在で 3 年次生であるとします。大学院入試は 7 ～ 8 月なので、そこか

回次	試験日	申込期間	結果発送日
第153回	3/14	1/5～2/2	4/13
第154回	5/30	3/1～4/20	6/29
第155回	6/27	5/6～5/18	7/27

ら逆算すると、今から受験出来るのは次の回です（一応自分の責任で調べてください）。

第 155 回の結果発送が、大学院の出願に間に合うかどうかは非常に微妙です。とりあえず全部申し込むことをおすすめしますが、第 153 ～ 154 回をターゲットにして勉強するのが良いでしょう。つまり、あと 3 ヶ月ほどで 730 点が取れれば合格とします。ゼロから始めるのであれば、あまり時間がありません。

（STEP2：出題のパターンを知ろう）

TOEIC の出題パターンは毎回決まっています。これは大学入試センター試験と通じるところがあります。要するに、出てきそうなパターンを全部覚えてしまえば、問題演習を繰り返すだけで点が取れるようになってしまうのです。

Part1（リスニング）写真描写問題 10 問

写真を見て、それに対する説明文の中から正しい物を一個選びます。まあ、どうでも良い事しか聞いてきません。肩ならしならぬ、耳ならしということでチョロッと終わります。しかし最近は難化傾向にあるとも言われているので、きっちり得点を取りたい分野です。

Part2（リスニング）応答問題 30 問

質問を聞いて、それに対する応答文の中から正しい物を一個選びます。3 択です。「正しい応答文」と言っても無数にあり得るので、逆に言えば明らかな間違いが二つ混ざっています。つまり、明らかに間違っている選択肢を排除していけば、自ずから答えに辿り着くセクションです。消去法をフル活用しましょう。

Part3（リスニング）会話問題 30 問

二人の会話を聞いて、その会話に関する設問に解答します。このあたりからだんだん難しくなってきます。設問・選択肢は問題用紙に印刷されているので、速読力も必要です。分からなかったらランダムに塗ってさっさと次に行きましょう。でないと間に合わなくなります。

Part4（リスニング）説明文問題 30 問

Part3 と似ていますが、今度は話者が 1 人です。どうでも良い独り語りを聞いて、どうでも良い質問に答えます。まさしく茶番という表現がふさわしいですが、ここは我慢してください。これ

でリスニングセクションは終わりになります。

Part5（リーディング）短文穴埋め問題 40 問

短文穴埋めというのは、いわゆる文法問題です。リスニングで疲れた頭に丁度良い程度の難易度です。要するに、そんなに難しくないということです。センター試験の文法問題より簡単だと思います。三単現の S とか、名詞か形容詞か副詞かとか、そういうレベルなので、脳トレみたいな感じで解いて下さい。もちろん知らないと解けないような熟語も出ますが、分からなかったらランダムに塗ってさっさと次に行くことが大事です。これは TOEIC 全体の鉄則だと思いますが、決して分からない問題に時間を使いすぎないで下さい。

Part6（リーディング）長文穴埋め問題 12 問

長文穴埋めと言いますが、実質的にこれも文法問題の続きです。頭から通しで読むのは時間の無駄なので、パラパラパラッと読む能力を身につけて下さい（これも、TOEIC 全体に言える事だと思います)。前後 2 センテンスくらいを読めば解ける問題が多いです。もちろん分からなかったら戻るべきですが、時間を強く意識してください。たった 12 問しか無いので、少しでも詰まったらランダムに塗って、Part7 のために時間を残すべきです。

Part7（リーディング）長文読解 48 問

個人的にはここが最大の難所だと思います。初めて受験する Non-Native の方は、必ずと言っていいほどここを全て解き切ることが出来ないでしょう。E-mail や張り紙などを読んで、それに対する設問に答えるという方式なのですが、とにかく時間が足りません。一個一個は長文というほどの長文ではないですが、それが大量に出題されるので結局時間が足りなくなります。集中力も削がれます。最初から解いていったとして、多くの人は最後の 10 問くらいで時間切れになります。ここが 700 ～ 800 点に乗るかどうかの分水嶺になっているような気がしてなりません。とにかく、パターン暗記です。出題パターンを徹底的に暗記して脊髄反射的に解答できるようになれば、800 点はおろか 900 点も夢ではないでしょう（自分はまだ、この境地には達していません……)。

まあこんな感じです。あとは自分で調べてください。TOEIC の過去問題は公開されていませんが、公式問題集というものが本屋さんで買えるので、それを見るのが一番です（これは後でも使うので、ぜひお買い求めください)。

（STEP3：勉強しよう）

出題パターンを知ったら勉強を始めましょう。英語の勉強というとまず単語暗記……という方も多いと思いますが、個人的に単語集は費用対効果が悪いように思うので、自分はやりませんでした（900 点を狙うなら必要だと思いますが)。再三申し上げてきた通り、とにかく TOEIC 対策はパターンの暗記がモノを言うと思いますので、それに特化した参考書をご紹介します。別に宣伝をしている訳ではないので、ぜひ書店で手に取って確かめてみることをおすすめします。とにかく、自分はこれをやったという紹介です。

TOEIC テスト出まくりキーフレーズ（コスモピア：高橋基治著）

これは非常によく出来た本だと思います。要するに例文集なのですが、これを暗記する事で TOEIC の全てのセクションに対応出来ます。TOEIC 好みの表現が散りばめられている良書です。

CD もついているので、ぜひこれを暗記してください。個人的には、単語集は昆虫の標本であるのに対して、例文集は生きている昆虫だと思います。実際にセンテンスという形で生きたニュアンスを学ぶのは、確実に力が付くと思うし、そっちのほうがむしろ楽です。

TOEIC テスト Part7 を 1 問 1 分で解けるようになる本（小学館：高橋基治著）

これも出まくりキーフレーズと同じ著者ですが、非常によく出来ています。時間の足りない Part7 の出題パターンを全て暗記するという主旨です。大学入試と同じで、出題パターン自体はそう大きく変わる事はありません。ちょこっと数値を変えただけの問題が頻出しています。そういう中で、「どこかで見た事ある……」という感覚が得られれば圧倒的に有利です。その感覚を得るための本です。同じシリーズで「Part3&4 を一気に 3 問解けるようになる本」というのも出ていますが、こっちは僕はやりませんでした（やるにこした事はありません）。

TOEICTEST 文法別問題集（講談社：石井辰哉著）

私のような文法オタクにはたまらない一冊ですが、いかんせん 780 問もあるので、時間無かったらやらなくても良いです。ただ、分厚い本ですが、思っているよりは早く終わります。そして、二回くらい繰り返せば確実に文法力がつきます。ここでは TOEIC 対策に特化した文法ドリルというのは貴重なので、紹介しました。

TOEIC テスト新公式問題集〈vol.1 ～ 4〉（国際ビジネスコミュニケーション協会：ETS 著）

こういう場所で ETS にお金を巻き上げられるのは非常に悔しいのですが、それでも公式問題集の利用価値は極めて高いと言わざるを得ません。1 冊につき 2 回分の模擬試験が収録されています。vol4 が最新なので、vol3 と vol4 を買えば充分でしょう。2 冊とも買っても、たった公式テスト一回分の値段です。パターン暗記をやりつつ、この 2 冊を 3 回くらい問題演習すれば、TOEIC の問題形式には慣れると思います。ちなみに vol1 と vol2 は少し古くて、傾向から外れているのでおすすめしません。いわば、暗記したパターンの実戦場です。この本は必須なので、ぜひとも買ってください。買ったら解いてください。解き終わったらちゃんと復習（30 分でいいので）してください。それだけで絶対に力は付きます。この公式問題集を解かずにテストを受けてしまうと、きっと本来の実力よりも 10 ～ 50 点ほど低いスコアになってしまいます。もちろん何度も受けていれば本来の実力に収斂（しゅうれん）していきますが、何しろ我々には時間もお金も無いのです！ TOEIC ごときにかけるコストは最小にすべきです。もちろん代用できる良い問題集があればそれでも構いませんが、僕の知る限りは公式問題に勝るものは売られていません（ご存知でしたら、こっそり教えてください……）。

これらを勉強して頂くことをおすすめ致します。これだけやれば 730 ～ 800 点を十分狙っていけると思います。

■最後に

個人的に、英語は目的ではなく手段だと思います。院試、就職活動、留学……どれも大事なイベントですが、英語力はそれを成功させるための道具に過ぎないと思っています。ですので、それにかける労力は最小にすべきです。そのツールの一つとして、今回は TOEIC を提案させていただきました。

これで説明は終わりになります。最後まで読んで頂きありがとうございました。何か分からない事がありましたら、sasaki at coins tsukuba ac jp 宛にメール頂ければお返事差し上げます。

アイドルと共に生きる(誕生日編)

文 編集部 ranha

論文と格闘しても一日に半ページも解読出来ず苛々。難しい、意味が分からない…。
証明が上手くいかずうつけになったり、書くプログラムは処理系や証明や型がどう。私は今、プログラムを書いているのかしら？それとも Logic をしている？いや数学…?ぐにゃぁ破滅。
最近、今自分が何をしているのか時に不安になる事があります。
一人ではもうどうにも…。そんな時に支えてくれるのは家族、友人または音楽だったり、歌…。

苛々している時には少しゆったりとした曲を聴いて落ち着いてみたり、イイ感じに攻めたい時にはアップテンポの曲を聴いたりします。歌の力で異星人を撃退したり出来るのかどうかは分かりませんが、少なくとも私は救われています。その意味では、去年の紅白歌合戦のテーマ「歌の力∞無限大」というのはそうだと言えます。

そしてそんな歌をうたう素晴らしいアイドル[*1] ！！そう、私達はアイドルによって毎日正気を保って生活を送る事が出来るのです！！人間こそが希望！！アイドルは希望！！アイドルはアイドルです！！ラブとか夢とか言ってる場合ではありません！アイドルを！アイドルこそ！！！

さて、アイドルにも誕生日があります。そこでファンとして勝手に祝おう！！はい！祝いますよーー！！というのがこの記事の主旨です。

筆者は「アイドルマスター」[*2] に傾倒しており、率先して某所でここ数ヶ月、毎月(つまりアイドルの誕生日が毎月ある)誕生日会と称してアイドルを崇め奉るようにしています。次ページの 2 点の写真はその時の様子です。

「人間として終わっている」「悲しい事だ」「リアルを捨てた」などという意見を想定していたのですが、そういう人達はもうちょっと感謝の心を持っても良いんじゃないですか？！
「いつもありがとうございます」という感謝の気持ちを祝う事で表すのです。歌い手が存在してそこで 1 つの歌となるのであり、その事を蔑ろにしてはいけません。月に一度の誕生日会に、人が集まってアイドルさんについて話したり、ワイワイ出来る事はとても素晴らしいことです。特にこのような催しを共に開ける人間が居るというのはいやはやなんと幸運な事か…。
アイドルでなくとも、自分のお気に入りのキャラクターを祝して、あれこれと思いを馳せる日が一年に一度あっても良いと思います。いやあるべきだと思います。

さて来る 1 月 21 日は、アイドルマスター的に言うと「四条貴音」[*3] さんの誕生日があります。丁度この文章を読んでいるプロデューサーの皆さんも勝手に祝いましょう。これで 21 日に twitter で「貴音さん誕生日会なう！！」という発言が多く見られる事を楽しみにしています。

---

*1 Immortal Defender of Legatee の事ではありません。いや...違わなくはないんですが…

*2 http://www.bandainamcogames.co.jp/cs/list/idolmaster/

*3 http://961pro.jp/

貴音さんの誕生日を祝う

2009 年 11 月 23 日撮影（星井美希さんの誕生日会）

2009 年 12 月 24 日撮影（萩原雪歩さんの誕生日会）

情報科学類誌

WORD

From College of Information Science

WORD は仕分けされませんでした号

発行者　　情報科学類長

編集長　　中　裕太郎

製作・編集　筑波大学情報学群
情報科学類 WORD 編集部
（第三エリア C 棟 212 号室）

2010 年 1 月　初版第一刷発行