

卒業論文

可視化されたプログラミング環境の構築

歌代 拓未

萩野服部合同研究室

環境情報学部4年 71141311

utayo@tom.sfc.keio.ac.jp

t11131tu@sfc.keio.ac.jp

2015. 01. 20

目次

第1章 序論

第2章 プログラミングに対する意識調査

- 2.1 概要
- 2.2 調査結果
- 2.3 プログラミングに対するモチベーション

第3章 ビジュアルプログラミングツール 「Visual-Poyogramming」について

- 3.1 概要f
- 3.2 使用方法
- 3.3 フィードバック

第4章 今後の課題・考察

付録A ソースコード

- A.1 index.html
- A.2 style.css
- A.3 code.js
- A.4 script.html

第1章

序論

現代において、情報技術は生活に欠かせない技術となっている。これまでは人力や原始的な構造で動いていた様々なものが情報技術により制御され、駆動している。あらゆるものをインターネットに接続するInternet of Thingsの考え方や、ユビキタスサービスが今後の世界にとって重要だとされている。近年ではICタグが様々な場所で使えるようになり、一枚のICカードで電車やバス等、全ての交通機関を利用できるような社会になりうる。このような情報技術の発展と浸透により、人間とITの距離はより接近する。現代ではまだサービスの意味合いが強いが、将来的に医療や生産等人類の発展に欠かせない技術とも関連していくことが予想される。[1]

その情報技術の発展にエンジニアの存在は欠かせず、専門的な技術を持ったプログラマたちが今の社会の基盤を創り上げたと言って問題はないだろう。それほど現代と情報技術は深く関わっている。しかし時代は進み、必要な人材も変化しているのではと私は考えた。まだ情報技術と深く関わっていない分野、言うなれば新たな市場を発見し、開拓できるような人材がこれからの発展に有用だと言えるのではないか。無論、基礎的な情報技術の革新は限界には未だ遠く、通信速度、処理速度、小型化、軽量化など、より高性能な技術に向けて開発が進んでいる。それは近年のPCやポータブルデバイスの進化からも見て取れる。しかし、これらの技術者はコアであるがゆえに他の分野との関わりが薄い。通信と電波等、近い分野ならその限りではないが、新たな分野の開拓に進むのは難しいだろう。またアイデアがあったとしても、それを実現できる環境ではない。

情報技術というものは学問の一つでありながら一つのツールであり、様々な分野に応用できる多くの可能性を持っている。第一次産業である農業を例にとってみる。まず生産段階で、ITを利用したセンサや機械による農作物の管理が可能である。降雨量や湿度から栄養、水分量を調節することができる。流通に関しても、現在Web上のサービスにより多くの作物が売買されている。ローカルな特産品と現地に行かずに情報を得られるインターネットは相性がよい。このように多くの分野にITを組み込むためには、様々な分野で仕事をする人間が、基礎的な情報技術、もしくは情報技術に対する一定の理解を持っていることが必要であるのではないか。

そこで、本研究では「プログラミングをより理解しやすくするにはどうするか」を主題として、プログラミングに対する意識調査、そしてビジュアルプログラミングツール「Visual-Poyogramming（以下VP）」の実装を行った。調査はプログラミング初学者が理解しづらい場面、箇所はどこか、そして学習意欲を削ぐ要因は何かについてのアンケートを実施した。またツールの作成に関しては、プログラムの構造をビジュアル化し、複雑なコーディングを可能にするよりはむしろアルゴリズム構築の基礎を学んでもらうことを目的とした。また、「可視化された環境でのプログラミング」が理解にどのような影響を及ぼすかに対する調査も兼ねている。

第2章ではプログラミングに対する理解と、モチベーションを重視したアンケートによる考察を行う。また、先行研究や実際に稼働している初学者向けのサービスと合わせ、どのような形式、特徴を持ったものが初学者に優しく理解しやすいのかも考慮する。

第3章では実装したビジュアルプログラミングツール「VP」の解説を行う。構成や実装した機能、また利用者によるフィードバックを述べる。

第4章では全体を総括して今後の課題を述べる。本ツール「VP」はプロトタイプであり、今後実装するツールに向けどのような機能、デザインが望ましいのかをこれからの情報教育の展望と合わせて考察する。

第2章

プログラミングに対する意識調査

2.1 概要

この章ではプログラミング初学者が実際にプログラミングをどう捉えているかについて調査し、問題解決に何が必要かを考える。調査はアンケートを実施し、プログラミングをどのレベルまで理解しているか、またプログラミングに対しどのような感情を感じているか、作業が停滞する原因は何かを調べた。

2.2 調査結果

まず、プログラミング技術に関するアンケートについて考える。項目は以下のとおりである。これらの質問について、自身があるかを回答してもらった。回答はできる、おそらくできる、不安がある、できないの4段階とした。またこの回答を数値化し、その平均値を合わせて記載しておく。なおこのアンケートの対象はSFCにおけるプログラミングの初学者である。プログラミングの経験が半年以下の学生が多くを占めており、彼らの学んでいる言語はJavascriptである。

	回答
1, 「変数について、人に説明できるか」	2.5
2, 「代入について、人に説明できるか」	2.75
3, 「条件分岐 (if文) について、人に説明できるか」	3.375
4, 「while文について、人に説明できるか」	2.25
5, 「for文について、人に説明できるか」	1.75
6, 「プログラミングの構造は簡単だと思うか」	2.0
7, 「プログラミングは好きか」	1.875

表 1

今回、質問の形式を「理解しているか」ではなく「人に説明できるか」とした。理由としては「できる」と「おそらくできる」の差別化である。ある程度自身の技術として持っていないと人に説明できるとは選びづらく、「理解している人」と「なんとなく理解している人」を区別することができると考えた。

結果としては変数、代入に関する回答が条件分岐に関する回答より自信がないということになった。これは変数、代入に不安がある人が多いのではなく、if文なら説明できるという回答の人が多かったためである。条件文に関する処理は他の部分に比べわかりやすいと感じるようである。それと反してwhile、for文の理解は浅い生徒が多かった。条件文はプログラムの上部から流れていく動作に反する動きをしないため理解しやすいが、繰り返しの処理が見えないのが分かりにくさの原因となっているのだろう。

また構造は難しいかという質問に対し、やはり難しいと感じる生徒が多かった。上記の結果と合わせると、やはり繰り返しの処理や複数のif文の組み合わせが難しいと感じられるようである。それぞれの構文への理解と、その組み合わせによる処理が可視化できれば解決に近づけると考えられる。

次にプログラミングに対する意識についてのアンケートについて考える。こちらは具体的な事例を回答として受け取った。

- 1, 「プログラミングにおいて複雑だと感じるのはどのような箇所、場合か」
- 2, 「プログラミングにおいて、何度も躓くのはどのような箇所、場合か」
- 3, 「プログラミングにおいて、
モチベーションが下がると感じるのはどのような箇所、場合か」

表2

以上3点について質問を実施した。これにより初学者がプログラミングに複雑さを感じている点が何かを考える。まず多かった意見が、「中括弧による構造構築の難しさ」だった。Javascriptはブロックの構築に中括弧（“{”, “}”）を用いており、中括弧で括弧することでその内部が一つのブロックとなっていることを表現している。プログラムは字下げスタイル（Indent Style）を使用することでブロックの構造を視覚的に確認できる形で記述する場合が多い。だがこれはあくまでユーザー側の工夫であり、プログラムとは無関係のものである。初学者はこの技術を知らず、字下げが不十分なコードを

書いてしまう。結果として構造が見辛くなり、ブロックの構造が崩壊する。具体例としては「中括弧のどちらかが多い、少ない」といったものとしてエラーが現れる。またそもそも入れ子構造そのものを知らずにプログラムを組んでいる生徒も存在し、何のために中括弧を使用しているのかわからないという意見もあった。

次に「スペルミス」についての意見があった。変数、関数を呼び出すときにスペルミスが発生し原因が別にあると感じる、もしくは原因が全く分からずに作業が停滞する。この理由として、エラーメッセージの意味がわからないという意見が多かった。コーディングに慣れていない初学者は多くの場所でミスをする。小規模でのテストという手法をとらず、複数のエラーの要因をまとめてテストすることが多く、当然一度のデバッグではエラーは除去しきれない。結果、「いくらやってもエラーが消えない。プログラミングのデバッグの作業は大変で面倒。」といった感情を抱いてしまう。初学者にとってエラー出力やコンソール画面は身近なものではなく、今回被験者が利用していたHTMLとJavascriptでは、エラーの発生したWebページは何も表示されない、もしくは動作しない静的なページとなってしまう。「成果が現れる、実感できるまでにデバッグ作業が多く、時間がかかる」のがモチベーションの低下と密接に関係があるようだ。

2.3 プログラミングに対するモチベーション

上記したアンケートの結果から、初学者のプログラミングに対する意識がどのようなものか、またそのモチベーションに関わる要因が何かを考察する。まず、今回の被験者はSFCにおけるプログラミング入門のための授業であり、ほぼ全生徒が履修する授業の履修者である。よって、プログラミングに興味があるかないかに関わらず履修している。学習意欲の高い人ではある程度までの障害によるモチベーションの低下がわかりにくいため、今回はモチベーションの少ない初学者を対象とした。

簡潔な結果としては、「出どころのわからないエラーによる長時間、複数回の作業の停滞」が最も学習意欲の低下に関わっていることが推測される。構文として誤っている箇所がピンポイントでわからないこと、一部の間違いでそれに関わる大部分のコードが動作しなくなること、それに伴う作業の延長が原因と思われる。プログラミングの経験者であればエラーメッセージから予想がつくものも、初学者はそのエラーの原因、コードの箇所がわからない。経験者の補助を受けずに独学でプログラミングを学ぶことは学習意欲の少ない学生からすれば苦難の多い作業となる。

記憶には短期記憶、長期記憶があることは知られているが、別の分類としてエピソード記憶、意味記憶、手続き型記憶に区別できるとされている。エピソード記憶とは、

過去の体験からくる具体的でユニークな記憶である。意味記憶は誰もが共通して持つことができる抽象的な知識の記憶である。「H₂Oは水である」等、知識に関する記憶であり、学習を通して得ることができる。最後に手続き型記憶とは、技能をいかに実行するかということに関する、行動と結果の関係の記憶である。車の運転やスポーツ等、自身の行動によって結果がどう影響するかを経験によって得ることができる。

無論プログラミングの作業には意味記憶は必要となる。知識として最低限のものを持っていなければその技術を使用した経験を得ることができないからである。しかしその先で重要になるのが手続き型記憶といえるだろう。プログラムは構築するだけでは不十分であり、実行して初めて結果が得られる。これは運転やスポーツと同様に、自身の操作による結果への影響であり、その意味ではスポーツの練習とプログラミングは同様の種類の知識を得ることができる。よって、それらと同様に実践しその経験から得た知識を用いてさらに実行する作業、いわゆる練習という作業によって、プログラミングの技術を高めることができる。しかし、学習と実践の繰り返しの際、重要になるのがモチベーションである。その繰り返しによるモチベーション低下を上回る必要性や楽しさなどの学習意欲の増大がなければ、学習と実践のループは終了する。現れる形としては飽きた、つまらない、必要性を感じない等が考えられるが、それらは等しくモチベーションの低下と括ることができるだろう。現代における学習は意欲について問題意識を持っているものが多くあり、近年のプログラミング教育に関するツールも例外ではない。[2][3]

先行するサービス、ツールには当ツールのようにマウス操作でのプログラミングが可能であるものがある。Google社による入門者向けプログラミングサイト「Blocky Games」やMITによるビジュアルプログラミングツール「Scratch」などが代表的である。特に前者のサイトは幾つかの形式のゲームがあり、プログラミングから遠いパズルゲームのようなものから実際に用意された関数をJavascriptから記述するゲームなど、一つ一つをクリアしていくことでアルゴリズムとは何かを学ぶことができる。UIとしては完成度が高く、プログラミングに理解があるものが使えばある程度複雑な動きをするプログラムを書くことができる。例として、真ん中ほどのレベルのゲームにタートルがある。亀を模したカーソルを前進、後退、左右に旋回させることで亀の通った道筋を図形として描画することができるゲームだ。これには関数が実装されており、関数名、変数名の入力のしにくさが少々気になるものの再帰を利用したフラクタル図形程度の難易度のものなら少ない手間ですべて記述することができる。私はこのサイトはアルゴリズム構築の勉強にとっても有用であると感じている。難易度の設定、段階的に機械語に近くなっていくゲームシステムといい意欲的な初学者（特に独学が難しいであろう中高生など）に勧めうるものだろう。

しかし、上記のBlocky Gamesを筆頭に現在使われている初学者向けのツールには一つ問題がある。それはそのツール内でアルゴリズムを構築できるようになっても、実際にコーディングはできないことである。（学習の速度は変化すると考えられる。）学習意欲と成果の関係は極めて密接であり、結果がでないものに熱中する人は極めて少ない。ゲームの形式をとることはその点で優れているが、そのゲームを一通りクリアしたとしてプログラミング技術自体が向上していなければ意味がない。学習の敷居を少し下げることができるが、それにとどまってしまう。これを踏まえ、Javascriptの学習入門ツールの理想を述べたい。それはHTMLとCSSを連携できるビジュアルプログラミングツールである。まずHTMLの記述も初学者にとっては容易ではない。Javascriptと同様に入れ子構造を持つため、同様の悩みが起きやすいと言える。HTMLの形成、idやclassによるCSS、Javascriptとの連携を視覚的に行うことで、利用者は複雑なコーディングをせずにJavascript最大の特徴であるWebページとの連携、つまり動きのあるWebページを構築することができる。実際のプログラミングと直接的に関係があるものを成果物として制作することで、言語自体への興味を持ってもらうことが期待できる。この点については第4章で追記する。

第3章

ビジュアルプログラミングツール 「Visual-Poyogrammingについて」

3.1 概要

本ツール「Visual-Poyogramming」（以下VP）は、マウス操作によるプログラミングを可能にし、初学者にアルゴリズム構築に基礎を学んでもらうことを目的として作成した。第2章において論述した「モチベーションを下げる要因」を取り除いたUser-Interfaceを特徴としている。なお実装に使用した言語はJavascriptであり、ツール内で作成されるコードも同様である。

一つはブロックの形成に関するものであり、中括弧の数の不一致のないプログラミング構成が可能である。これはブロックを形成する文（while文, if文）に他の文を挿入、またブロック外へ他の文を出力することで実現されている。

一つは変数リストの実装である。これにより、利用者は一度宣言した変数をドラッグ&ドロップを用いて使用することができる。キーボードで変数名を入力することなく代入文や条件文の記述が可能であるため、スペルミスが起こらない仕様となっている。

3.2 使用方法

実装した機能と使用方法の説明を行う。

3.2.1 メインウィンドウ

全体の画面（図1）から、大まかな機能の説明をする。詳しい機能や画面遷移は後述する。まず、①が書いたコードが現れる領域である。②が変数、定数を扱う領域であり、このエリアからドラッグすることで入力が可能となる。③がドラッグされた値を

受け取る領域であり、代入文や条件文を作成することができる。④の領域では、①に制御文を代入することができる。

3.2.2 コードエリア

図1の①にあたる部分である。②、③により作成された代入文や④により挿入される制御文をブロック構造で可視化する。図2はVPにより実際に書かれたFizzBuzz問題の一般的な解答、図3はその解答を実際のJavascriptで書いたコードである。入れ子構造をブロックによる内包と色分けによって表現している。

また行番号が書いて有る箇所をクリックすることでその行を選択することができる。他の選択状態でif文、while文をマウスオーバーしている時、図4のようにExchange、Insertの二つのボタンが表示される。行が選択された状態で他の行をクリック、もしくはExchangeのボタンをクリックすることで行の入れ替えが可能である。なお、ブロックを構成する文とそのブロック内の行は入れ替えができないため、アラートが発生する。もう一つのInsertのボタンを押すことで、その文がもつブロック内に選択された行を挿入することができる。ブロック内の行が選択されている場合、InsertのボタンはExportボタンとなり、その行をブロック内から取り出すことができる。

3.2.3 変数・定数エリア

図1における②にあたる部分である。「Variables」には変数が、「Other」には定数が挿入できるボタンが設置されている。(図5、図6を参照)これらのボタンから図1における③、アサインエリアへドラッグ&ドロップすることで挿入が可能である。変数は図4における追加ボタンを押すことで変数名の入力画面へと移り、同じ変数名の変数が宣言されていなければ新しい変数として図5に描画される。またOtherのNumber、Stringについては、アサインエリアへドロップされた時にそれぞれ実数、文字列の入力画面へと移る。その画面で正しく入力されると挿入が行われる。

3.2.4 アサインエリア

図1における③の部分である。②の変数定数エリアから値をドラッグして、このアサインエリアで代入文、条件文を作成することができる。灰色の「Null」と記述されているボックス、またすでに代入されている青色のボックスへドロップすることで、ドラッ

グ元の値を代入することができる。この操作を使い文を構築する。まず代入文を作成するAssignエリア（図7）について解説する。

図中にTargetとあるが、これが代入される変数である。通常のコードではただ左辺として扱われるが、初学者にはそもそも代入という操作の理解が不十分なことがある。よって、代入先と代入される内容を分割することで構文への理解を深める狙いからこのようなデザインとした。Targetの下、灰色の枠線で囲まれた部分があるが、これが代入される内容を記述するエリアである。変数リストと同じく「+」と書かれた追加のボタンがあるが、このボタンがクリックされることにより演算子と新たなNullボックスのセットが追加される。演算子は「+、-、×、÷、%」である。

Target右側の三つのボタンについて説明する。最も右側のゴミ箱のアイコンがある。利用者はこのアイコンへドラッグすることで変数、定数の削除が行える。削除された変数のボックス内にはNullボックスが設置される。またその左側、リロードのアイコンがある。このアイコンではフィールドのリセットを行える。Targetの内容もゴミ箱へドラッグすることで削除できるが、まとめて消去したい時にこのボタンを使うことができる。またその左側、折れた矢印のアイコンがある。これは挿入ボタンであり、これを使うことで実際に代入を行う。なおエリア内に未定義のボックスがあるとアラートが発生する。プログラマはしばしばnull値を代入するが、初学者向けではないと判断しこの仕様とした。

次にConditionエリア（図8）について説明する。ここでは条件文を挿入することができる。Assignエリアと同様にTargetが存在するが、こちらはドラッグではなく実際に代入する条件文をクリックすることで図のように表示される。図7の例は1行目のif文という意味となる。またこのエリアにおける追加ボタンだが、これは条件の追加が可能である。今回は実装の都合により、最大二項による条件設定となっている。図の通り、and（&&）とor（||）の論理演算子を使用することができる。

最後に④のその他のエリアについて説明する。まずNewのページにあるボタン、New IfとNew Whileのボタンだが、これにより初期状態のif文、while文をコードエリアに挿入することができる。またOtherのページにはMake Codeのボタンがある。このボタンは新規ウィンドウを作成し、コードエリアにVPにより描画されているコードをJavascriptの形式で出力する。

3.3 フィードバック

今回実装したプログラムを初学者に見て、使ってもらい、そのフィードバックとして今後役に立つものがあった。一つは機能の欠落についてである。この度の実装の仕様により、Javascriptの基本構文で実装されていない部分がある。具体例としてはfor文、複数の条件による条件文、null値やundefined値の挿入、評価、関数の定義、利用等である。これより高度なものは初学者向けではないとして意図的に省いたものだが、これらの機能は実装するために十分に時間を割けなかった。

一つは繰り返しの可視化についてである。アンケートによる調査でも現れていたが、「条件分岐はわかるが繰り返しの処理が複雑でわからない」という学生が多かった。テキスト上ではただの文字列であり、その中で戻る、また他の行へジャンプするという処理が慣れないようである。コーディングの可視化にとどまらず、生成されたコードに対するビジュアル化にも着手してほしいという意見が多くあった。本ツールの目的として、「可視化された環境において作業することでアルゴリズム構築の基礎を学ぶ」というものがあったが、コーディングだけでは不十分であり、制作物がアルゴリズムとしてどのように動いているのかを小さいスケールで視覚的に理解することが重要であるとわかった。問題に対する答え合わせを行うことで、学生の自信と理解を生むのだろう。人間の理解のメカニズムとの関連付けも合わせて、今後の課題としたい。

上記のように指摘をもらったが、良かった点としてブロック構造が挙げられる。アンケートにあった「中括弧の数が合わずにエラーが出る」という意見、また「ブロック構造を理解していないため、そもそもなぜ中括弧でくるのかわからない」という意見を構造を可視化し、さらにその形でコーディングすることで解決した。しかし当初の狙いであった「ビジュアル化された環境でのプログラミング」を可能にするためには未実装の機能が多く、このツールだけを渡して実際にプログラミングをしてもらう作業は助言が必要となり、生徒のみによる学習はできなかった。何が問題かを一部生徒に質問したところ、「そもそも慣れないプログラミングであり、プロトタイプ、未完成であるプログラムによるコーディングが難しい」との意見を複数もらった。

これにより必要だと思われる事項が二つある。一つはその意見どおりに機能実装である。限られた機能でプログラムを書くということはプログラミングの学習の上で必要なことではあるが、上級のプログラマがトレーニングとして用いることもあり、場合によってはより困難な作業となる。よってまず初学者が必要とするであろう機能を

一通り実装することが不可欠であると考えられる。もう一つは学習用のトレーニングページの作成である。一通り教わったとはいえ、理解が不十分な生徒を多く対象としていたので、VPを使用する前に基礎的な構文を教える時間を設けていた。より学習ツールとして完成させるためには、自由なコーディングを可能にするとともに一つずつ問題をクリアしていくシステムを作ることでより良い学習環境となる。算数、数学の教科において計算法や方式の説明の後に練習問題があり、それをクリアすることで理解を深めるように、各構文に対しその基礎をデモすることでその後の問題が活きるだろう。上述した完成したコードのビジュアライズと合わせ実装することで、高い効果を期待できる。具体的にはコードが処理される順番を描画する、ステップ実行を可視化しデータフローと合わせて描画する等の例が考えられる。

また、学生へ「どのようなプログラミングツール、エディタ、システムがあったらプログラミングがすきになるか」という質問をした。意見の一つは、「日本語化してほしい」というものだった。あまりなじみのないアルファベットが理解できない法則で並んでいることが学生をプログラミングから遠ざけているようだ。これについてはまず構造を理解することが重要だと考えた。不規則に並んでいるように見えるためモチベーションが下がっているとすれば、まずその構造、プログラミング言語がどのような成り立ちをしているのかを理解することでこの問題は解決できる。

一つはコードに対する可視化である。上述したように、完成物に対し理解が追いつかないと「よく理解できないが動いている」という状況になる。このような学習を重ねることで、学生がその場しのぎのコピーアンドペーストによるコーディングを繰り返すことが予想できる。そのため、基礎的な学習において「学習意欲の減退が少ない」、かつ「理解しやすい」方法が必要である。

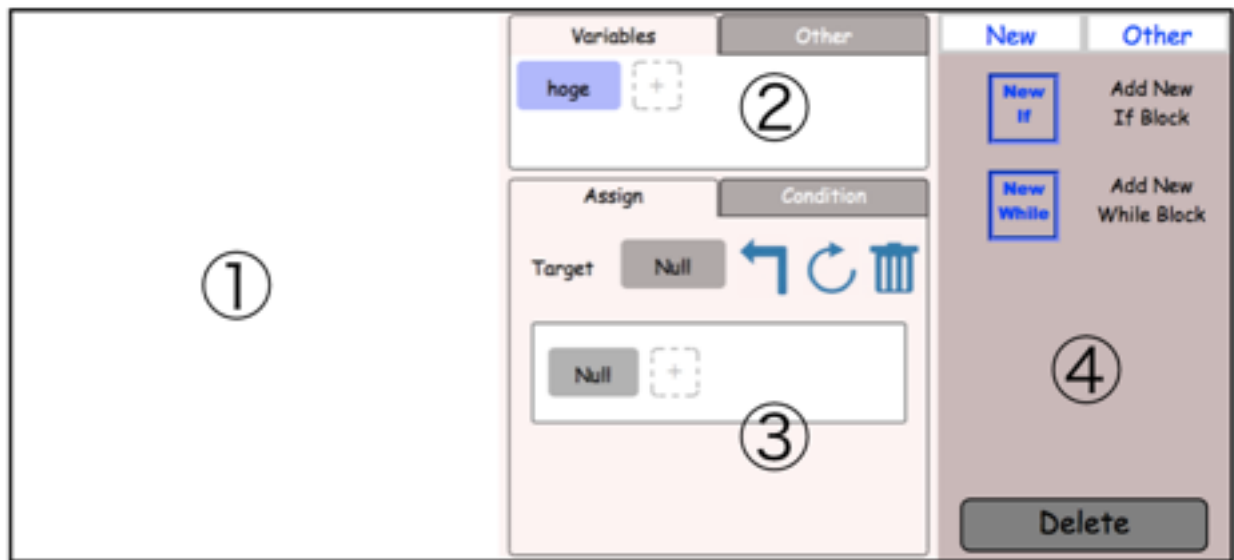


図 1

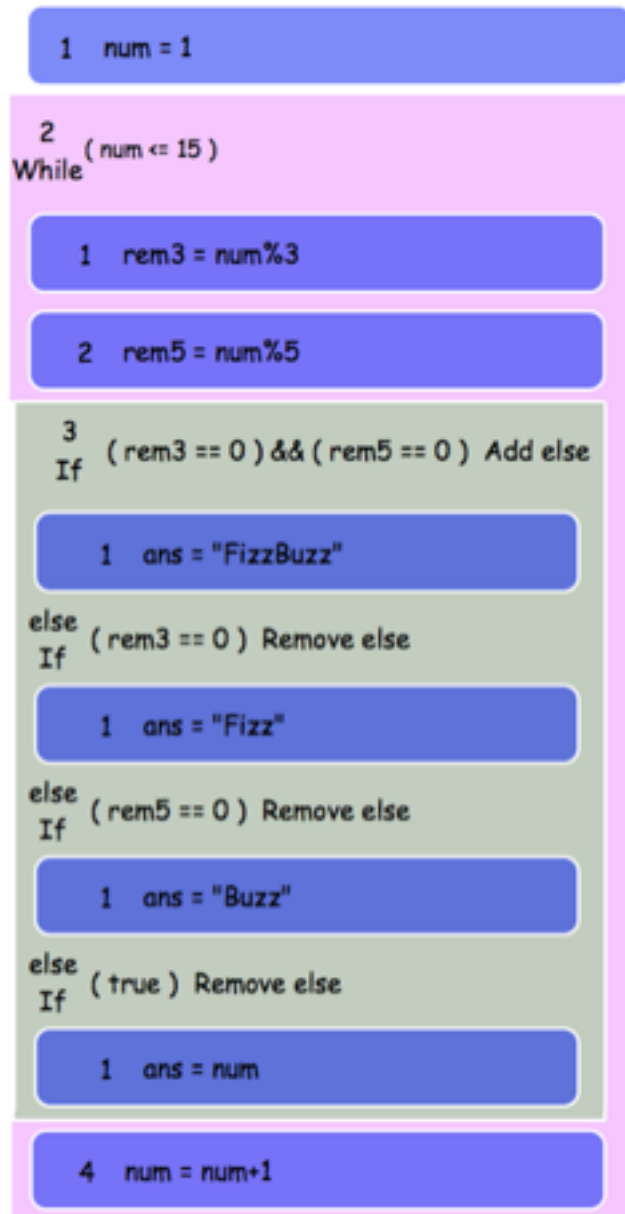


図 2

```

num = 1;
while (num<=15) {
    rem3 = num % 3;
    rem5 = num % 5;

    if((rem3==0) && (rem5==0)){
        ans = "FizzBuzz";
    }else if(rem3==0){
        ans = "Fizz";
    }else if(rem5==0){
        ans = "Buzz";
    }else if(true){
        ans = num;
    }

    num = num + 1;
}

```

図 3



図 4



図 5



図 6

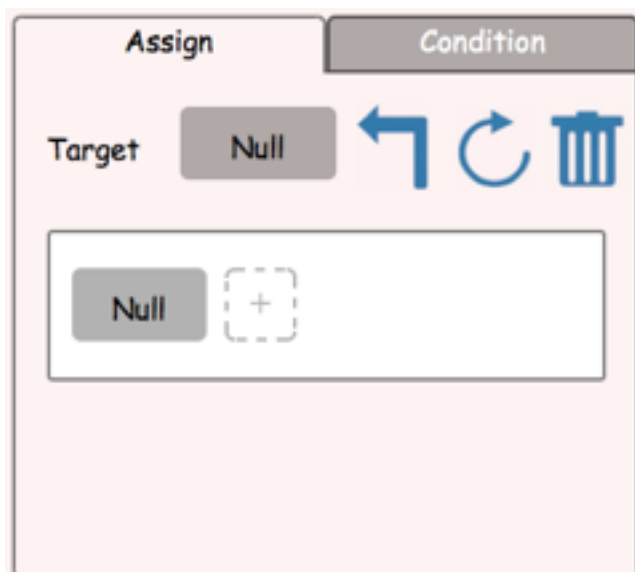


図 7

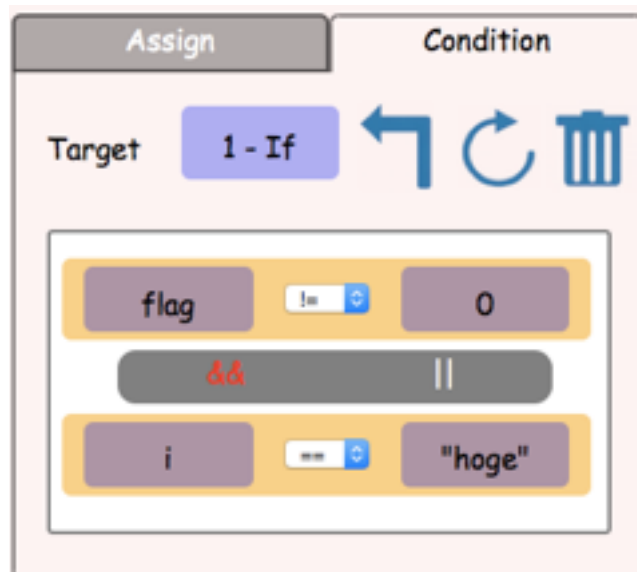


図 8

第4章

今後の課題・考察

今回の実装から明らかになった課題をまとめる。まず一つ目は機能面の実装である。実装されていない基礎的な機能が幾つかある。具体的にはfor文、余り（%）を使用した条件文の設定、関数関連等が挙げられる。満足にコードをかけないエディタでは初学者向けとは言い難い。この点が最優先で解決すべき事項であろう。

次に、その先に何をすべきかを考察する。被験者の対応からブロック構造のビジュアライズに関しては概ね良好だと感じたが、条件分岐と繰り返しの構造を理解するのに複雑さが存在するようだ。今回の主題は「可視化されたプログラミング環境」というものだったが、それと合わせて「完成したプログラムの可視化」が必要だろう。実際に書いたコードがどのような順序で動き、宣言した変数の値がどのように変遷しているのかを可視化することで理解が深まることが予想できる。自身のコードがどのように動いているかを認識するだけでなく、基本的な構文の理解や確認、サンプルコードの構成を解析することもできる。

さらに、今後の展望を述べる。2.3にも記述したように、Javascriptの特徴に合った成果物を作成することで言語への理解が深まることが予想できる。ではそれを実現するためにどのような機能が必要かを考える。まずJavascriptの特徴はHTML、CSSとの連携による動的なWebページの作成が挙げられる。クライアントサイドのコードとして記述することでサーバを利用することもできるが、初期段階としてはユーザーの操作等のアクションに応じて形を変えるページが基礎となる。その制作のステップとしては以下のような例があげられる。

- 1, HTMLの記述：動かすコンテンツの作成。
- 2, CSSの記述：コンテンツの配置。
- 3, Javascriptの記述：コンテンツを動かすためのスクリプトの記述。
- 4, 関連付け：HTML要素とスクリプトの関連付け。

様々な手法でWebページは構築されるが、多くの場合このようなステップを通していると考えられる。1、2のステップは構造自体は簡単であり、ページの構築自体は難しい。しかしページとスクリプトを連携させることで手間は増加する。イベントドリブンの理解が必須となり、それを知らずに初学者は苦しんでいる。変数にHTML要素を代入するという動作も難しいようである。HTMLとJavascriptのつながりを視覚的に行う、またそのつながりを可視化することで、どのように連携しているかの理解を深めることができるだろう。

謝辞

本研究の機会を与えてくださった、慶應義塾大学 環境情報学部 服部隆志教授、萩野達也教授に深く感謝いたします。

また助言をいただくと共に授業のアシスタントとして採用していただいた、慶應義塾大学 環境情報学部 清水智公講師に感謝いたします。初学者向けの授業におけるアシスタントとしての経験は、本研究に大きな意欲と意義を与えてくださったと感じています。

参考文献

- [1] 小林 健一郎、佐野 典秀 他 「教養情報科学概論 -ユビキタス時代の情報活用-」
共立出版 2004 年
- [2] 今栄 国春 「新版 教育の情報化と認知科学」 福村出版 1998 年
- [3] 岡本 浩一 「スランプ克服の法則」 PHP出版 2004 年

付録A

ソースコード

A.1 index.html

```
<!DOCTYPE html>
<head>
  <meta charset="UTF-8">
  <title>Visual-Poyogramming</title>
  <script type="text/javascript" src="code.js"></script>
  <link href="style.css" rel="stylesheet" type="text/css">
</head>

<body>
  <div id="project">
    <div id="coding_area">
      <div id="struct_area">
        <!--コードエリア/-->
      </div>

      <div id="desk_area">
        <div id="variable_list_area">
          <div class="as_select">
            <div class="as_bar" id="var_list_bar">Variables</div>
            <div class="as_bar as_unselected " id="other_list_bar">Other</div>
          </div>

          <div id="variable_list" class="list_area">
            <!--変数エリア/-->
          </div>

          <div id="other_list" class="list_area">
            <!--定数エリア/-->
            <div class="other_element" id="other_num">Number</div>
            <div class="other_element" id="other_str">String</div>
          </div>
        </div>

        <div id="edit_area">
          <div class="type_select">
            <div id="type_assign" class="type_selected">Assign</div>
            <div id="type_condition" class="type_unselected">Condition</div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
```

```

<div class="edit_box" id="edit_assign">
  <!--Assignエリア/-->
  <div style="width: 100%;height: 5%"></div>
  <p style="float: left;margin-left: 5%;margin-right: 5%;">Target </p>
  <div class="v_null" id="target">Null</div>
  </img>
  </img>
  </img>
  <div id="assign_area">
  </div>
</div>

<div class="edit_box" id="edit_condition">
  <!--Conditionエリア/-->
  <div style="width: 100%;height: 5%"></div>
  <p style="float: left;margin-left: 5%;margin-right: 5%;">Target </p>
  <div class="v_null" id="cond_target">Null</div>
  </img>
  </img>
  </img>
  <div id="condition_area">
  </div>
</div>
</div>
</div>
</div>

<div id="tool_bar" style="display:block">
  <!--ツールバー/-->
  <div id="tool_bar_header">
    <a class="tool_link" onclick="tool_bar_switch('new')">
      <div class="tool_bar_button">New</div>
    </a>
    <a class="tool_link" onclick="tool_bar_switch('other')">
      <div class="tool_bar_button">Other</div>
    </a>
  </div>

  <div id="tool_bar_item">
    <div id="tool_bar_new">
      <div id="new_if" class="tool_area">
        <!--ifブロック作成/-->
        <div class="tool_blank"></div>
        <div class="tool_box_area">
          <a class="tool_link">
            <div class="tool_box" onclick="tb_new_if()">
              New</br>
              If
            </div>
          </a>
        </div>
        <div class="tool_text">
          Add New</br>
          If Block
        </div>
        <div class="tool_input">
          <input type="button" value="Make New If-Block"

```

```

        id="if_button" onclick="make_new_if()">>
    </div>
    <div class="tool_blank"></div>
</div>

<div id="new_while" class="tool_area">
<!--whileブロック作成/-->
    <div class="tool_blank"></div>
    <div class="tool_box_area">
        <a class="tool_link">
            <div class="tool_box" onclick="tb_new_while()">
                New</br>
                While
            </div>
        </a>
    </div>
    <div class="tool_text">
        Add New</br>
        While Block
    </div>
    <div class="tool_input">
        <input type="button" value="Make New While-Block"
            id="while_button" onclick="make_new_while()">
    </div>
    <div class="tool_blank"></div>
</div>

<div id="tool_bar_other">
    <div id="make_code_button" class="tool_area">
        <!--コード出力/-->
        <div class="tool_blank"></div>
        <div class="tool_box_area">
            <a class="tool_link">
                <div class="tool_box" onclick="make_code()">
                    Make</br>
                    Code
                </div>
            </a>
        </div>
        <div class="tool_text">
            Show Code</br>
            (New Window)
        </div>
        <div class="tool_blank"></div>
    </div>
</div>

</div>
<div id="tool_bar_delete">
    <!--行の削除/-->
    <div id="delete_button" class="stop" onclick="delete_line()">
        Delete
    </div>
</div>
</div>

```

```
</div>  
</body>  
</html>
```


A.2 style.css

```
#project{
    /* ウィンドウ全体 */
    background-color: rgba(255,0,0,0.1);
    width: 1000px;
    height: 450px;
    border-style: solid;

    font-family: "Comic Sans MS",fantasy;
}

#project:hover{
    cursor: default;
}

#coding_area{
    background-color: rgba(255,255,255,0.5);
    width: 760px;
    height: 450px;
    float: left;
    user-select: none;
    -moz-user-select: none;
}

#option_bar{
    background-color: black;
    width: 756px;
    height: 26px;
    border-style: solid;
    border-color: white;
    border-width: 2px;
}

.option{
    float: left;
    width: 60px;
    text-align: center;
}

.option:hover{
    cursor: pointer;
}

.option_link{
    color: white;
    user-select: none;
    -moz-user-select: none;
}

#struct_area{
    /* コードエリア */
    background-color: white;
    width: 400px;
```

```

        height: 100%;
        float: left;
        overflow: scroll;
    }

    #desk_area{
        width: 360px;
        height: 450px;
        float: left;
    }

    .as_select{
        width: 100%;
        height: 25%;
    }

    .as_bar{
        float: left;
        width: 50%;
        height: 100%;
        line-height: 30px;
        text-align: center;
        border-left: 2px solid rgba(0,0,0,0.5);
        border-right: 2px solid rgba(0,0,0,0.5);
        border-top: 2px solid rgba(0,0,0,0.5);
        border-radius: 5px 5px 0 0 / 5px 5px 0 0;
        box-sizing: border-box;
    }

    .as_unselected{
        color: white;
        background-color: rgba(0,0,0,0.3);
        border-bottom: 2px solid rgba(0,0,0,0.5);
    }

    #variable_list_area{
        float: left;
        margin-top: 2px;
        width: 96%;
        height: 30%;
        margin-left: 2%;
    }

    .trash{
        float: right;
        margin-right: 0px;
        border: 2px solid transparent;
        border-radius: 5px;
    }

    .restart{
        float: right;
        margin-top: 3px;
        margin-right: 0px;
        border: 2px solid transparent;
        border-radius: 5px;
    }

```

```

.insert{
    float: right;
    margin-right: 10px;
    border: 2px solid transparent;
    border-radius: 5px;
}

.cond_trash{
    float: right;
    margin-right: 0px;
    border: 2px solid transparent;
    border-radius: 5px;
}

.cond_restart{
    float: right;
    margin-top: 3px;
    margin-right: 0px;
    border: 2px solid transparent;
    border-radius: 5px;
}

.cond_insert{
    float: right;
    margin-right: 10px;
    border: 2px solid transparent;
    border-radius: 5px;
}

#v_title{
    margin: 0;
    font-size: 16px;
    line-height: 20px;
    text-align: center;
    height: 25%;
}

.list_area{
    float: left;
    background-color: white;
    overflow-y: scroll;
    overflow-x: visible;
    width: 100%;
    height: 70%;
    border-left: 2px solid rgba(0,0,0,0.5);
    border-right: 2px solid rgba(0,0,0,0.5);
    border-bottom: 2px solid rgba(0,0,0,0.5);
    box-sizing: border-box;
    border-radius: 0 0 5px 5px / 0 0 5px 5px;
}

#variable_list{
    display: block;
}

#other_list{

```

```

        display: none;
    }

    .other_element{
        float: left;
        width: 48%;
        height: 62%;
        font-size: 20px;
        text-align: center;
        line-height: 52px;
        box-sizing: border-box;
        margin: 1%;
        margin-top: 15px;
        background-color: rgba(0,255,0,0.5);
        border-radius: 5px;
        border: 2px solid transparent;
    }

    #other_num:hover{
        border: 2px solid red;
    }

    #other_str:hover{
        border: 2px solid red;
    }

    .v_element{
        height: 40px;
        min-width: 25%;
        float: left;
        background-color: rgba(0,0,255,0.3);
        line-height: 30px;
        padding: 5px;
        border-radius: 5px;
        border: 2px solid rgba(0,0,0,0);
        margin: 5px;
        font-size: 16px;
        text-align: center;
        box-sizing: border-box;
        overflow: scroll;
    }

    .v_element:hover{
        border: 2px solid red;
    }

    .v_selected{
        height: 40px;
        min-width: 25%;
        float: left;
        background-color: rgba(255,0,0,0.7);
        line-height: 30px;
        padding: 5px;
        border-radius: 5px;
        border: 2px solid rgba(0,0,0,0);
        margin: 5px;
        font-size: 16px;
    }

```

```

        text-align: center;
        box-sizing: border-box;
    }

    .v_null{
        height: 40px;
        min-width: 25%;
        float: left;
        background-color: rgba(0,0,0,0.3);
        line-height: 30px;
        padding: 5px;
        border-radius: 5px;
        border: 2px solid transparent;
        margin: 5px;
        font-size: 16px;
        text-align: center;
        box-sizing: border-box;
    }

    .v_add{
        height: 40px;
        min-width: 40px;
        float: left;
        background-color: white;
        color: rgba(0,0,0,0.3);
        line-height: 25px;
        padding: 5px;
        border-radius: 5px;
        border: 2px dashed rgba(0,0,0,0.3);
        margin: 5px;
        font-size: 16px;
        text-align: center;
        box-sizing: border-box;
    }

    .v_add:hover{
        color: red;
    }

    .v_opr{
        float: left;
        height: 100%;
        width: 15%;
        line-height: 40px;
        margin-left: 3px;
        margin-right: 3px;
    }

    .target_over:hover{
        border: 2px solid red;
    }

    #edit_condition #cond_target:hover{
        border: 2px solid transparent;
    }

    #assign_area{

```

```

        box-sizing: border-box;
        padding-top: 4%;
        padding-left: 2%;
        border: 2px solid rgba(0,0,0,0.5);
        border-radius: 3px;
        float: left;
        margin-top: 5%;
        margin-left: 5%;
        min-width: 90%;
        height: 30%;
        background-color: white;
        overflow-x: scroll;
        display: flex;
        flex-direction: row;
    }

    #edit_area{
        float: left;
        margin-left: 2%;

        text-align: center;
        width: 96%;
        height: 70%;
    }

    #condition_area{
        box-sizing: border-box;
        padding-top: 4%;
        padding-left: 2%;
        float: left;
        margin-top: 5%;
        margin-left: 5%;
        border: 2px solid rgba(0,0,0,0.5);
        border-radius: 3px;
        width: 90%;
        height: 60%;
        background-color: white;
        overflow-y: scroll;
        text-align: center;
    }

    #condition_area .v_null,
    #condition_area .v_element,
    #condition_area .v_selected{
        height: 80%;
        width: 80%;
        line-height: 25px;
        margin-left: 10%;
        margin-top: 4%;
    }

    .condition_box{
        float: left;
        width: 97%;
        height: 30%;
        margin-bottom: 5px;
        background-color: rgba(255,165,0,0.5);

```

```

        border-radius: 5px;
    }

.c_element_div{
    float: left;
    width: 40%;
    height: 100%;
}

.c_null{
    height: 40px;
    width: 80px;
    float: left;
    background-color: rgba(0,0,0,0.3);
    line-height: 20px;
    padding: 10px;
    border-radius: 5px;
    border: 2px solid transparent;
    margin: 10px;
    font-size: 16px;
    text-align: center;
    box-sizing: border-box;
}

.c_opr{
    float: left;
    width: 20%;
    height: 80%;
}

.c_select{
    float: center;
    line-height: 40px;
}

.c_add{
    float: left;
    height: 30px;
    width: 40%;
    background-color: white;
    color: rgba(0,0,0,0.3);
    line-height: 25px;
    border-radius: 5px;
    border: 2px dashed rgba(0,0,0,0.3);
    margin-left: 30%;
    margin-top: 5px;
    margin-bottom: 5px;
    font-size: 16px;
    text-align: center;
    box-sizing: border-box;
}

.c_add:hover{
    color: red;
}

.c_logOpr{

```

```

        float: left;
        height: 30px;
        width: 80%;
        background-color: gray;
        color: white;
        border-radius: 10px;
        margin-left: 10%;
        margin-top: 0px;
        margin-bottom: 5px;
    }

    .c_logElement{
        float: left;
        height: 90%;
        width: 50%;
    }

    .c_logSelected{
        color: red;
    }

    .type_select{
        width: 100%;
        height: 10%;
    }

    .edit_box{
        float: left;
        width: 99%;
        height: 88%;
        border-radius: 0 0 5px 5px / 0 0 5px 5px;
        border-bottom: 2px solid rgba(0,0,0,0.5);
        border-right: 2px solid rgba(0,0,0,0.5);
        border-left: 2px solid rgba(0,0,0,0.5);
    }

    #edit_assign{
        display: block;
    }

    #edit_condition{
        display: none;
    }

    .type_selected{
        float:left;
        text-align: center;
        width: 50%;
        height: 100%;
        box-sizing: border-box;
        border-radius: 5px 5px 0 0/5px 5px 0 0;
        border-top: 2px solid rgba(0,0,0,0.5);
        border-right: 2px solid rgba(0,0,0,0.5);
        border-left: 2px solid rgba(0,0,0,0.5);
    }

```



```

.type_unselected{
    float:left;
    text-align: center;
    width: 50%;
    height: 100%;
    background-color: rgba(0,0,0,0.3);
    color: white;
    box-sizing: border-box;
    border-radius: 5px 5px 0 0 / 5px 5px 0 0;
    border-top: 2px solid rgba(0,0,0,0.5);
    border-right: 2px solid rgba(0,0,0,0.5);
    border-left: 2px solid rgba(0,0,0,0.5);
    border-bottom: 2px solid rgba(0,0,0,0.5);
}

.line{
    background-color: rgba(0,0,255,0.5) !important;
    float: left;
    height: 46px;
    width: 92%;
    border-style: solid;
    border-color: white;
    border-width: 2px;
    border-radius: 10px;
    margin-left: 3%;
    margin-top: 5px;
    margin-bottom: 5px;
    user-select: none;
    -moz-user-select: none;
}

.line_selected{
    background-color: rgba(255,0,0,0.5);
    float: left;
    height: 46px;
    width: 92%;
    border-radius: 10px;
    margin-left: 3%;
    margin-top: 5px;
    margin-bottom: 5px;
    border-style: solid;
    border-color: white;
    border-width: 2px;
    user-select: none;
    -moz-user-select: none;
}

.line_blank{
    height: 46px;
    width: 10px;
    float: left;
}

.line_number{
    cursor: move;
    float: left;
    height: 100%;

```

```

        width: 46px;

        text-align: center;
        line-height: 48px;
    }

    .line_struct{
        line-height: 46px;
        float: left;
        user-select: none;
        -moz-user-select: none;
    }

    .if_option{
        width: 100px;
        line-height: 46px;
        float: right;
        text-align: center;
        user-select: none;
        -moz-user-select: none;

        display: none;
    }

    .if_option:hover{
        color: red;
    }

    .if_rest{
        max-width: 300px;
        height: 100%;
        float: right;
    }

    .box{
        float: left;
        width: 95%;

        border-style: solid;

        border-color: white;
        border-width: 2px;
        overflow: hidden;
        user-select: none;
        -moz-user-select: none;
    }

    .box_selected{
        background-color: rgba(0,255,0,0.25);
        float: left;
        width: 95%;
        border-style: solid;

        border-color: red;
        border-width: 2px;
        position: relative;
        overflow: hidden;
    }

```

```

        user-select: none;
        -moz-user-select: none;
    }

    .box_blank{
        background-color: rgba(0,255,0,0);
        border-style: solid;
        border-color: white;
        border-top-color: ;
    }

    .if{
        height: 46px;
        width: 100%;
        padding-top: 10px;
        padding-bottom: 10px;
        user-select: none;
        -moz-user-select: none;
    }

    .add_else{
        float: left;
        height: 46px;
        text-align: center;
        line-height: 46px;
        margin-left: 10px;
    }

    .add_else:hover{
        color: white;
    }

    .while{
        height: 46px;
        width: 100%;
        padding-top: 10px;
        padding-bottom: 10px;
        user-select: none;
        -moz-user-select: none;
    }

    .if_number{
        cursor: move;
        float: left;
        height: 100%;
        width: 46px;

        text-align: center;
        line-height: 23px;
    }

    .if_selected{
        background-color: rgba(0,255,0,0.2);
        height: 46px;
        width: 99%;
        border-style: solid;
        border-color: red;
    }

```

```

        border-width: 2px;
        user-select: none;
        -moz-user-select: none;
    }

    .if_struct{
        line-height: 46px;
        float: left;
        user-select: none;
        -moz-user-select: none;
    }

    .if_struct_selected{
        line-height: 46px;
        float: left;
        color: red;
        user-select: none;
        -moz-user-select: none;
    }

    .if_struct:hover{
        cursor: pointer;
        color: white;
    }

    .if_struct_selected:hover{
        cursor: pointer;
    }

    #tool_bar{
        width: 240px;
        height: 450px;
        float: right;
        background-color: rgba(0,0,0,0.2);

        display: none;
    }

    #tool_bar_header{
        width: 240px;
        height: 40px;
        user-select: none;
        -moz-user-select: none;
    }

    #tool_bar_item{
        width: 240px;
        height: 360px;
    }

    #tool_bar_delete{
        width: 240px;
        height: 50px;
    }

    #delete_button{
        margin: 0 auto;

```

```

        font-size: 24px;
        text-align: center;
        width: 200px;
        height: 40px;
        border-radius: 8px;
    }

    .stop{
        color: black;
        background-color: gray;
        border: 2px solid black;
        cursor: default;
    }

    .active{
        color: black;
        background-color: red;
        border: 2px solid black;
        cursor: pointer;
    }

    .active:hover{
        color: white;
        border: 2px solid white;
        cursor: pointer;
    }

    #tool_bar_new{
        display: block;
    }

    #tool_bar_assign{
        display: none;
    }

    #tool_bar_other{
        display: none;
    }

    .tool_bar_button{
        width: 50%;
        height: 35px;
        float: left;
        border-style: solid;
        border-width: 4px;
        border-color: rgba(0,0,0,0.2);
        background-color:white;
        text-align: center;
        font-size:20px;
        box-sizing: border-box;
    }

    .tool_bar_button:hover{
        cursor: pointer;
    }

    .in_toolbar_text{

```

```

        user-select: none;
        float: center;
        text-align: center;
    }

    .tool_blank{
        width: 240px;
        height: 10px;
        float:left;
    }

    .tool_box_area{
        width:100px;
        height:60px;
        float:left;
    }

    .tool_box{
        width:50px;
        height:50px;
        border-style:groove;
        border-width:5px;
        font-family:"Arial Black";
        font-size: 14px;
        float:right;
        text-align: center;
        user-select: none;
        -moz-user-select: none;
    }

    .tool_box:hover{
        cursor: pointer;
    }

    .tool_text{
        width:130px;
        height:60px;
        float:right;
        text-align: center;
        display: block;
        user-select: none;
        -moz-user-select: none;
        cursor: default;
    }

    .tool_input{
        width:130px;
        height:60px;
        float:right;
        text-align: center;
        font-size: 16px;
        display: none;
    }

    .assign_selector{
        width:130px;
        height:60px;

```

```

float:right;
text-align: center;
font-size: 16px;
display: none;
}

#if_button{
width: 90%;
height: 80%;
text-align: center;
float: center;
}

#while_button{
width: 90%;
height: 80%;
text-align: center;
float: center;
}

#assign_var_selector{
font-size: 14px;
width: 120px;
}

.tool_link{
color:#0000ff;
}

.tool_link:hover{
color:#ff0000;
}

```

A.3 code.js

```
var project;
var tool_bar_opened = "new";
var selected_line_num;
var selected_var_name;
var selected_struct_num;
var dragged_area;
var dragged_fixed_value;
var serial_number;
var add_div;

var lines = [];
var variables;

var line = function(_number,_kind,_name,_value){
    this.number = _number;
    this.kind = _kind;
    if(_kind=="Var"){
        this.name = _name;
        this.var_value = new variable(_name);
        variables[_name] = this.var_value;
    }else if(_kind=="Assign"){
        this.name = _name;
        if(_value){
            this.var_value = variables[_name];
            this.assign_value = _value;
            this.type = null;
            if(_value!="BEFORE_ASSIGN")
                this.struct = new struct(_value);
        }else{
            console.log("please insert assign value...");
            window.alert("please insert assign value...");
        }
    }else if(_kind=="If"||_kind=="While"){
        this.condition = "( true )";
        this.inner_lines = [];
        if(_kind=="If"){
            this.else_array = [];
            this.else_array[0] = new line(0,"Start");
        }
    }else if(_kind=="Else"){
        this.condition = "( true )";
        this.inner_lines = [];
    }
}

var struct = function(value){
    //      文字列を演算子で分割し、各要素に分けて保存する
    var contents = [];
    var str0 = [];
    var str = String(value);
    str = str.split("+");
    for(var prop in str){
        if(prop!=0)
```



```

        contents[contents.length] = new operator("+");
    str[prop] = str[prop].split("-");
    for(var prop0 in str[prop]){
        if(prop0!=0){
            contents[contents.length] = new operator("-");
        }
        contents[contents.length] = new content(str[prop][prop0]);
        str0[str0.length] = str[prop][prop0];
    }
}
this.contents = contents;
}

var content = function(value){
    var elements = [];
    var str = value.split("**");
    var str0 = [];
    for(var prop in str){
        if(prop!=0)
            elements[elements.length] = new operator("**");
        str[prop] = str[prop].split("/");
        for(var prop0 in str[prop]){
            if(prop0!=0)
                elements[elements.length] = new operator("/");
            elements[elements.length] = new element(str[prop][prop0]);
        }
    }
    this.elements = elements;
}

var element = function(value){
    this.element = value;
}

var operator = function(kind){
    this.operator = kind;
}

var variable = function(_name){
    this.name = _name;
    this.value = null;
}

var add_new_line = function(kind,name,value,parentIf){
    var l = lines.length;
    if(name){
        if(!variables[name]){
            if(kind=="Var"){
                lines[l] = new line(serial_number,kind,name,null);

                document.getElementById("new_var_input").value= "";
                serial_number++;
                add_all_lineView(add_div,lines);
            }
        }else{
            if(kind=="Assign"){
                lines[l] = new line(serial_number,kind,name,value);

```

```

        serial_number++;
        add_all_lineView(add_div,lines);
    }else{
        window.alert(name + " is already exist...");
        console.log(name+" is already exist...");
    }
}
}else{
    if(parentIf){
        parentIf.else_array[parentIf.else_array.length] = new line(serial_number,"If");
        serial_number++;
    }else if(kind=="If"||kind=="While"){
        if(selected_line_num==1){
            lines[] = new line(serial_number,kind,null,null);
            lines[].inner_lines[0] = new line(0,"Start");
        }else{
            var parent = search_line(selected_line_num).inner_lines;
            parent[parent.length] = new line(serial_number,kind,null,null);
            parent[parent.length-1].inner_lines[0] = new line(0,"Start");
        }
        serial_number++;
        add_all_lineView(add_div,lines);
    }else{
        window.alert("Please insert variable name...");
        console.log("please insert variable name...");
    }
}
selected_line_num = -1;
}

var add_all_varView = function(){
    var hoge = document.getElementById("variable_list");
    while(hoge.firstChild)
        hoge.removeChild(hoge.firstChild);

    for(var prop in variables){
        add_new_varView(variables[prop].name);
    }
    vList_addButton();
}

var add_new_varView = function(name,area){
    if(!area){
        area = document.getElementById("variable_list");
    }

    var new_div = document.createElement("div");
    new_div.className = "v_element";
    new_div.innerHTML = name;
    new_div.id = name;

    area.appendChild(new_div);
    new_div.addEventListener("mousedown", function(){drag_start(name,new_div)},false);
}

var drag_classChange = function(div){

```

```

if(div)
    div.className = "v_selected";
var tar = document.getElementById("target");
tar.classList.add("target_over");

var eList = document.querySelectorAll(".v_element");
for(var prop=0;prop<eList.length;prop++){
    eList[prop].classList.add("target_over");
}

var nList = document.querySelectorAll(".v_null");
for(var prop=0;prop<nList.length;prop++){
    nList[prop].classList.add("target_over");
}

var trash = document.querySelector(".trash");
trash.classList.add("target_over");
var cTrash = document.querySelector(".cond_trash");
cTrash.classList.add("target_over");
}

var drag_start = function(name,div){
    console.log("Dragging : "+name);
    console.log(div);
    drag_classChange(div);

    selected_var_name = name;
    dragged_area = div;
}

var num_drag_start = function(){
    drag_classChange();
    console.log("Dragging :Number")

    dragged_area = "number";
}

var str_drag_start = function(){
    drag_classChange();
    console.log("Dragging :String");

    dragged_area = "string";
}

var drop = function(area){
    var new_null;

    //      ドラッグされている変数があり、それがvariable_listから来ていない場合。
    //      ドラッグ元のdivを元の状態（Null）に戻す。
    if(selected_var_name){
        if(dragged_area.parentNode.id!="variable_list"){
            new_null = document.createElement("div");
            if(dragged_area.id=="target")
                new_null.id = "target";
            new_null.className = "v_null";
            new_null.innerHTML = "Nullaaa";

```

```

        new_null.addEventListener("mouseup",function(){drop(new_null)},false);
        var parent = dragged_area.parentNode;

        parent.insertBefore(new_null, dragged_area);
        parent.removeChild(dragged_area);
    }
}

if(dragged_area=="number"){
    if(area.id=="target")
        window.alert("変数をいれてください。");
    else
        num_drop_window(area);
}else if(dragged_area=="string"){
    if(area.id=="target")
        window.alert("変数をいれてください。");
    else
        str_drop_window(area);
}else{
    if(!selected_var_name)
        return false;

    var sv;
    if(variables[selected_var_name]){
        sv = variables[selected_var_name].name;
    }else if(selected_var_name!="Null"){
        sv = selected_var_name;
    }
    console.log(area);
    if(area.id=="target"){
        var tar = document.getElementById("target");
        tar.className = "v_element";
        tar.innerHTML = sv;
        tar.addEventListener("mousedown", function(){drag_start(sv,tar)},false);
    }else if(area){
        if(!area.parentNode)
            area = new_null;
        area.className = "v_element";
        area.innerHTML = sv;
        area.addEventListener("mousedown", function(){drag_start(sv,area);},false);
    }
}
dragged_area = null;
selected_var_name = null;
}

var window_drop = function(){
    var sel = document.querySelector(".v_selected");
    if(sel){
        sel.className = "v_element";
        document.getElementById("target").classList.remove("target_over");
    }

    var eList = document.querySelectorAll(".v_element");
    for(var prop=0;prop<eList.length;prop++){
        if(eList[prop].classList.contains("target_over"))
            eList[prop].classList.remove("target_over");
    }
}

```

```

var nList = document.querySelectorAll(".v_null");
for(var prop=0;prop<nList.length;prop++){
    nList[prop].classList.remove("target_over");
}

var trash = document.querySelector(".trash")
trash.classList.remove("target_over");
var cTrash = document.querySelector(".cond_trash")
cTrash.classList.remove("target_over");
}

var num_drop_window = function(area){
    var num = window.prompt("実数を入力してください。");
    if(num===null){
        window.alert("キャンセルされました。");
    }else{
        num = parseFloat(num);
        if(isNaN(num)){
            window.alert("実数でないと判断されました。");
        }else{
            console.log(area);
            if(area.id=="target")
                area = document.getElementById("target");
            area.className = "v_element";
            area.innerHTML = num;
            area.addEventListener("mousedown", function(){drag_start(num,area)},false);
        }
    }
}

var str_drop_window = function(area){
    var str = window.prompt("文字列を入力してください。");
    if(str==false){
        window.alert("キャンセルされました。");
        return false;
    }
    if(area.id=="target")
        area = document.getElementById("target");
    area.className = "v_element";
    area.innerHTML = "\"" + str + "\"";
    area.addEventListener("mousedown", function(){drag_start(str,area)},false);
}

var trash_drop = function(){
    var root = dragged_area;
    if(root.parentNode.id=="variable_list"){
        console.log("Delete Variable: "+selected_var_name);
        delete variables[selected_var_name];
        add_all_varView();
    }else{
        var new_null = document.createElement("div");
        if(dragged_area.id=="target")
            new_null.id = "target";
        new_null.className = "v_null";
        new_null.innerHTML = "Null";
    }
}

```

```

        new_null.addEventListener("mouseup",function(){drop(new_null)},false);
        var parent = dragged_area.parentNode;

        parent.insertBefore(new_null, dragged_area);
        parent.removeChild(dragged_area);
    }
    selected_var_name = null;
}

var add_nullset = function(){
    var area = document.getElementById("assign_area");
    if(area.querySelector(".v_add"))
        area.removeChild(area.querySelector(".v_add"));
    add_opr();
    add_null();
    add_addButton();
}

var add_null = function(area){
    if(!area)
        area = document.getElementById("assign_area");

    var div = document.createElement("div");
    div.className = "v_null";
    div.innerHTML = "Null";
    div.addEventListener("mouseup",function(){drop(div)},false);

    area.appendChild(div);
}

var add_addButton = function(area){
    if(!area)
        area = document.getElementById("assign_area");

    var div = document.createElement("div");
    div.className = "v_add";
    div.innerHTML = "+"
    div.title = "Add";
    div.addEventListener("click",function(){add_nullset()},false);

    area.appendChild(div);
}

var assign_restart_window = function(){
    var flag = window.confirm("エリアをリセットします。");
    if(flag)
        assign_restart();
}

var assign_restart = function(){
    var target = document.getElementById("target");
    var new_null = document.createElement("div");
    new_null.id = "target";
    new_null.className = "v_null";
    new_null.innerHTML = "Null";
    new_null.addEventListener("mouseup",function(){drop(new_null)},false);
    var parent = target.parentNode;

```

```

parent.insertBefore(new_null, target);
parent.removeChild(target);

var area = document.getElementById("assign_area");
while(area.firstChild)
    area.removeChild(area.firstChild);

add_null();
add_addButton();
selected_var_name = null;
dragged_area = null;
}

var assign_export_window = function(){
    if(!assign_export_check()){
        window.alert("未定義の項目があります。");
        return false;
    }

    if(window.confirm("代入を行います。")){
        assign_export();
    }
}

var assign_export_check = function(){
    var target_area = document.getElementById("target");
    var aList = document.getElementById("assign_area");
    if(target_area.className=="v_null"||aList.querySelector(".v_null"))
        return false;

    return true;
}

var assign_export = function(){
    var target_area = document.getElementById("target");
    var target;
    target = target_area.innerHTML;
    var aList = document.getElementById("assign_area").children;
    var hoge = "";
    for(var prop in aList){
        var div = aList[prop];
        if(div.className=="v_element")
            hoge += div.innerHTML;
        else if(div.className=="v_opr"){
            var sel = div.querySelector("select");
            hoge+= sel.options[sel.selectedIndex].value;
        }
    }
    assign_restart();
    add_new_line("Assign", target, hoge);
}

var cond_restart_window = function(){
    var flag = window.confirm("エリアをリセットします。");
    if(flag)
        cond_restart();
}

```

```

}

var cond_restart = function(){
    var target = document.getElementById("cond_target");
    var new_cNull = document.createElement("div");
    new_cNull.className = "v_null";
    new_cNull.id = "cond_target";
    new_cNull.innerHTML = "Null";

    var parent = target.parentNode;
    parent.insertBefore(new_cNull, target);
    parent.removeChild(target);

    var area = document.getElementById("condition_area");

    while(area.firstChild)
        area.removeChild(area.firstChild);

    make_cBox();
    make_cAdd();
}

var cond_insert_window = function(){
    var div = document.querySelector(".if_struct_selected");
    var num = div.parentNode.querySelector(".if_number");
    num = num.innerHTML.replace("<br>", "行目の");
    var condArea = document.querySelector("#condition_area");
    var fNull = condArea.querySelector("v_null");
    if(!div){
        window.alert("条件を設定する箇所を設定してください。");
    }else if(fNull){
        window.alert("未定義の箇所があります。");
    }else{
        var str = div.innerHTML;
        var flag = window.confirm(num+"に代入します。");
        if(flag)
            cond_insert(div);
        else
            window.alert("キャンセルされました。");
    }
}

var cond_insert = function(div){
    var str = "( ";
    var title = "";
    var condArea = document.querySelector("#condition_area");
    var boxList = condArea.querySelectorAll(".condition_box");

    for(var i=0;i<boxList.length;i++){
        var box = boxList[i];
        console.log(box);
        if(i!=0){
            str += ") "
            var logOpr = condArea.querySelector(".c_logSelected").innerHTML;
            str += logOpr + " ( ";
        }
        var eList = box.querySelectorAll(".v_element");

```



```

        str += eList[0].innerHTML + " ";
        str += box.querySelector("select").value + " ";
        str += eList[1].innerHTML + " ";
    }
    str += ") ";

    div.className = "if_struct";
    var hoge = search_line(selected_struct_num);
    selected_struct_num = -1;
    hoge.condition = str;
    add_all_lineView(add_div, lines);

    cond_restart();
    console.log(str);
}

var vList_addButton = function(){
    var area = document.getElementById("variable_list");

    var div = document.createElement("div");
    div.className = "v_add";
    div.innerHTML = "+";
    div.title = "変数を追加"
    div.addEventListener("click", function(){vList_addWindow();}, false);

    area.appendChild(div);
}

var vList_addWindow = function(){
    var name = window.prompt("新規変数名を入力してください");
    if(name==""){
        vList_addWindow();
        return false;
    }else if(variables[name]){
        window.alert(name+" は既に登録されています。");
        return false;
    }else if(name){
        vList_add(name);
        return true;
    }
    return false;
}

var add_opr = function(area){
    if(!area)
        area = document.getElementById("assign_area");

    var opr = ["+", "-", "*", "/", "%"];
    var opr_value = ["+", "-", "x", "÷", "%"]

    var div = document.createElement("div");
    div.className = "v_opr";
    var sel = document.createElement("select");
    for(var prop in opr){
        var hoge = document.createElement("option");
        hoge.value = opr[prop];
        hoge.innerHTML = opr_value[prop];
    }

```

```

        sel.appendChild(hoge);
    }

    div.appendChild(sel);
    area.appendChild(div);
}

var vList_add = function(name){
    variables[name] = new variable(name);
    add_all_varView();
}

var select_struct = function(serial,line,area){
    var target = document.querySelector("#cond_target");
    console.log("hoge");
    if(selected_struct_num==serial){
        //選択されているstructがクリックされた場合。deselect.
        area.className = "if_struct";
        selected_struct_num = -1;

        var div = document.createElement("div");
        div.innerHTML = "Nullweeei";
        div.className = "v_null";
        div.id = "cond_target";
        var p = target.parentNode;
        p.insertBefore(div, target);
        p.removeChild(target);

        type_change("Assign");
    }else{
        //cond_targetが未選択の場合。insert
        //選択されているstructとクリックされたstructが違う場合。overwrite.
        var hoge = document.querySelector(".if_struct_selected");
        if(hoge)
            hoge.className = "if_struct";
        console.log(area);

        selected_struct_num = serial;
        area.className = "if_struct_selected";

        var n = area.parentNode.querySelector(".if_number").innerHTML;
        n = n.replace("<br>"," - ");
        target.innerHTML = n;
        target.className = "v_element";
        type_change("Condition");
    }
}

var make_cBox = function(){
    var area = document.getElementById("condition_area");
    var selValue = ["<", ">", "<=", ">=", "==", "!="]

    var box = document.createElement("div");
    box.className = "condition_box";

    var make_cElementDiv = function(){
        var cElementDiv = document.createElement("div");

```

```

        cElementDiv.className = "c_element_div";
        var cNull = document.createElement("div");
        cNull.className = "v_null";
        cNull.innerHTML = "Null";
        cNull.addEventListener("mouseup",function(){drop(cNull)},false);
        cElementDiv.appendChild(cNull);
        return cElementDiv;
    }

    var ce0 = new make_cElementDiv();
    var ce1 = new make_cElementDiv();

    var cOpt = document.createElement("div");
    cOpt.className = "c_opr";
    var cSel = document.createElement("div");
    cSel.className = "c_select";
    var cSelector = document.createElement("select");
    for(var prop in selValue){
        var opt = document.createElement("option");
        opt.innerHTML = selValue[prop];
        opt.value = selValue[prop];
        cSelector.appendChild(opt);
    }
    cSel.appendChild(cSelector);
    cOpt.appendChild(cSel);

    box.appendChild(ce0);
    box.appendChild(cOpt);
    box.appendChild(ce1);

    area.appendChild(box);
}

var make_cSet = function(){
    var area = document.getElementById("condition_area");
    var button = area.querySelector(".c_add");
    area.removeChild(button);

    make_cLogOpr();
    make_cBox();
}

var make_cAdd = function(){
    var area = document.getElementById("condition_area");

    var cAdd = document.createElement("div");
    cAdd.className = "c_add";
    cAdd.innerHTML = "+";
    cAdd.addEventListener("click", make_cSet, false);

    area.appendChild(cAdd);
}

var make_cLogOpr = function(){
    var area = document.getElementById("condition_area");
    var cLogOpr = document.createElement("div");
    cLogOpr.className = "c_logOpr";

```

```

var and = document.createElement("div");
and.className = "c_logElement c_logSelected";
and.title = "かつ";
and.innerHTML = "&&";
and.addEventListener("click", function(){change_cLogOpr(and)}, false);

var or = document.createElement("div");
or.className = "c_logElement";
or.title = "または";
or.innerHTML = "||";
or.addEventListener("click", function(){change_cLogOpr(or)}, false);

cLogOpr.appendChild(and);
cLogOpr.appendChild(or);

area.appendChild(cLogOpr);
}

var change_cLogOpr = function(area){
    var parent = area.parentNode;
    var hoge = parent.querySelector(".c_logSelected");
    if(!hoge){
        console.log("Error...");
        return false;
    }

    hoge.classList.remove("c_logSelected");
    area.classList.add("c_logSelected");
}

var test_assign = function(){
    console.log("test assign function");
    var v = document.f.assign_value.value;
    var n = document.getElementById("selector").value;
    assign(n,v);
}

var assign = function(name,value){
    console.log(name + " = " + value + ",");
    add_new_line("Assign",name,value);
}

var add_expression = function(){
    var v = document.f.assign_value;
    var opr = document.getElementById("opr").value;
    var n = document.getElementById("selector").value;
    v.value += opr + n;
}

var tbn = function(name){
    return "tool_bar_"+name;
}

var tool_bar_visible = function(){
    var toolBar = document.getElementById("tool_bar");
    var v;

```

```

        if(toolBar.style.display=="block") v="none";
        else v="block";
        toolBar.style.display=v;
        document.getElementById(tbn(tool_bar_opened)).style.display=v;
    }

var tool_bar_switch = function(name){
    if(name==tool_bar_opened){
        console.log(name+" is already opened...");
        return 0;
    }else if(name=="new"){
        console.log("Tool Bar Switch for New");
    }else if(name=="assign"){
        console.log("Tool Bar Switch for Assign");
    }else if(name=="other"){
        console.log("Tool Bar Switch for Other");
    }else{
        console.log("Error::"+name+" is not Exist...");
        return -1;
    }
    document.getElementById("tool_bar_"+name).style.display="block";
    document.getElementById("tool_bar_"+tool_bar_opened).style.display="none";
    tool_bar_opened = name;
}

var type_change = function(area_name){
    var as = document.getElementById("edit_assign");
    var co = document.getElementById("edit_condition");

    if(area_name=="Assign"){
        as.style.display = "block";
        co.style.display = "none";
        document.getElementById("type_assign").className = "type_selected";
        document.getElementById("type_condition").className = "type_unselected";
    }else if(area_name=="Condition"){
        as.style.display = "none";
        co.style.display = "block";
        document.getElementById("type_assign").className = "type_unselected";
        document.getElementById("type_condition").className = "type_selected";
    }
}

var tpChange0 = function(area_name){
    var as0 = document.getElementById("variable_list");
    var co0 = document.getElementById("other_list");

    if(area_name=="varList"){
        as0.style.display = "block";
        co0.style.display = "none";
        document.getElementById("var_list_bar").className = "as_bar";
        document.getElementById("other_list_bar").className = "as_bar as_unselected";
    }else if(area_name=="otherList"){
        as0.style.display = "none";
        co0.style.display = "block";
        document.getElementById("var_list_bar").className = "as_bar as_unselected";
        document.getElementById("other_list_bar").className = "as_bar";
    }
}

```

```

}

var add_all_lineView = function(area,list,layer,elseCount){

    if(area.className=="if"||area.className=="while")
        area = area.parentNode;

    if(!elseCount)
        elseCount = 0;

    var structArea = area;
    var children = structArea.childNodes;

    if(area.id=="struct_area"){
        while(structArea.firstChild){
            structArea.removeChild(structArea.firstChild);
        }
        add_all_varView();
    }

    if(!layer)var layer = 0;

    for(var prop in list){
        if(list[prop]){
            serial = list[prop].number;
            if(list[prop].kind=="Var"){
                //console.log("::"+list[prop].name);
                add_new_lineView(serial,prop,"NEW_VARIABLE",list[prop].name,area,list,layer);
            }else if(list[prop].kind=="Assign"){
                //console.log(list[prop].value);
                add_new_lineView(serial,prop,"ASSIGN_VARIABLE",list[prop].name,area,list,layer);
            }else if(list[prop].kind=="If"||list[prop].kind=="While"){
                if(elseCount!=0)
                    add_new_boxView(serial,prop,area,layer,prop);
                else
                    add_new_boxView(serial,prop,area,layer,elseCount);
                var search_class;
                if(list[prop].kind=="If")
                    search_class = ".if";
                else if (list[prop].kind=="While")
                    search_class = ".while";
                var hoge = area.querySelectorAll(search_class);
                hoge = hoge[hoge.length-1];
                console.log(hoge);

                add_all_lineView(hoge,list[prop].inner_lines,layer+1);
                if(list[prop].else_array&&list[prop].else_array.length!=1){
                    add_all_lineView(hoge,list[prop].else_array,layer,elseCount+1);
                }
            }
        }
    }
}

var add_new_lineView = function(serial,number,value,name,area,parent,layer){
    var struct = area;

```

```

var line = document.createElement("div");
line.className = "line";

for(var i=0;i<layer;i++){
    var blank = document.createElement("div");
    blank.className = "line_blank";
    line.appendChild(blank);
}

var lineNum = document.createElement("div");
lineNum.innerHTML = number;
lineNum.className = "line_number";
line.appendChild(lineNum);
var lineStruct = document.createElement("div");
lineStruct.className = "line_struct";

var str;

if(value=="NEW_VARIABLE"){
    str = " : new variable";
    all_add_selector(name);        //test code;
}else if(parent[number].assign_value=="BEFORE_ASSIGN"){
    str = " = Empty!";
}else if(value=="ASSIGN_VARIABLE"){
    if(parent[number].type=="String")
        str = " = \"\" + parent[number].assign_value + "\" : Assign Fixed String";
    else if(parent[number].type=="Number")
        str = " = " + parent[number].assign_value + " : Assign Fixed Number";
    else
        str = " = " + parent[number].assign_value;
}

str = name + str;
lineStruct.innerHTML = str;
line.appendChild(lineStruct);

struct.appendChild(line);

line.addEventListener("click", function(){select_line(serial,name,line)},false);
}

var add_new_boxView = function(serial,number,area,layer,elseCount){
    var l = search_line(serial);
    var box;
    if(elseCount==0){
        box = document.createElement("div");
        box.className = "box";
    }else{
        box = document.querySelectorAll(".box");
        box = box[box.length-1];
    }

    var struct = document.getElementById("struct_area");
    var line = document.createElement("div");
    console.log(l);
    if(l.kind=="If"){

```

```

        line.className = "if";
        box.style.backgroundColor = "rgba(0,255,0,0.25)"
    }
    else if(l.kind=="While"){
        line.className = "while";
        box.style.backgroundColor = "rgba(255,0,255,0.25)"
    }

    var rest_width = 300;

    for(var i=0;i<layer;i++){
        var blank = document.createElement("div");
        blank.className = "line_blank";
        line.appendChild(blank);
        rest_width -= 10;
    }

    var lineNum = document.createElement("div");
    if(elseCount==0)
        lineNum.innerHTML = number + "<br>" + l.kind;
    else
        lineNum.innerHTML = "else<br>if";
    lineNum.className = "if_number";
    line.appendChild(lineNum);
    var lineStruct = document.createElement("div");
    lineStruct.className = "if_struct";
    lineStruct.innerHTML = l.condition;
    console.log(l.condition);
    var change_div = document.createElement("div");
    change_div.className = "if_option";
    change_div.innerHTML = "Exchange";
    var insert_div = document.createElement("div");
    insert_div.className = "if_option if_insert";
    insert_div.innerHTML = "Insert";

    line.appendChild(lineStruct);
    line.appendChild(change_div);
    line.appendChild(insert_div);

    if(l.kind=="If"){
        //      else と else if を追加するボタンを作る
        if(elseCount==0){
            var addElse = document.createElement("div");
            addElse.innerHTML = "Add else";
            addElse.className = "add_else";
            addElse.addEventListener("click", function(){add_else(serial)}, false);

            line.appendChild(addElse);
        }else{
            var removeElse = document.createElement("div");
            removeElse.innerHTML = "Remove else";
            removeElse.className = "add_else";
            removeElse.addEventListener("click", function(){remove_else(serial,elseCount)},false);

            line.appendChild(removeElse);
        }
    }
}

```



```

box.appendChild(line);

if(elseCount==0)
    area.appendChild(box);

change_div.addEventListener("click", function(){select_line(serial,name,line)},false);
insert_div.addEventListener("click", function(){insert_line(serial,line)},false);
lineNum.addEventListener("click", function(){select_line(serial,name,line)},false);
line.addEventListener("mouseover", function(){if_option_view(serial,line,true)},false);
line.addEventListener("mouseout", function(){if_option_view(serial,line,false)},false);

lineStruct.addEventListener("click", function(){select_struct(serial,line,lineStruct)}, false);
}

var add_else = function(serial){
    var l = search_line(serial);
    console.log(l.else_array.length);
    add_new_line("If",null,null,l);
    add_all_lineView(add_div,lines);
}

var remove_else = function(serial,count){
    var l = search_line(serial,0,lines,null,"ParentBox");
    console.log(count);
    l = l.splice(count,1);
    add_all_lineView(add_div,lines);
}

var all_add_selector = function(){
    // セレクターを全削除してからvariablesの中身を追加
    var selector = document.getElementById("assign_var_selector");
    while(selector.firstChild){
        selector.removeChild(selector.firstChild);
    }

    for(var prop in variables){
        var new_option = document.createElement("option");
        new_option.innerHTML = variables[prop].name;
        new_option.setAttribute("value",variables[prop].name);
        selector.appendChild(new_option);
    }
}

var add_selector = function(name){
    var selector = document.getElementById("assign_var_selector");
    add_option(selector,name);
}

var add_option = function(node,option){
    var new_option = document.createElement("option");
    new_option.innerHTML = option;
    new_option.setAttribute("value",option);
    node.appendChild(new_option);
}

var add_test_line = function(){

```

```

var name = document.f.new_var.value;
add_new_line('Var',name,null);
}

var select_line = function(number,name,line_area){
    if(selected_line_num==number){
        console.log(number + " is deselected.");
        selected_line_num = -1;
        var sel = search_line(number);
        if(sel.kind=="If"||sel.kind=="While"){
            line_area.parentNode.className = "box";
        }else{
            line_area.className = "line";
        }
        tool_bar_switch('new');
        delete_button_change('stop');
    }else{
        if(selected_line_num==--1){
            //          ラインが選択されていない状態
            var sl = search_line(number,0);
            if(sl){
                if(sl.kind=="If"||sl.kind=="While"){
                    console.log(number + " is " + sl.kind);
                    line_area.parentNode.className = "box_selected";
                    line_area = line_area.parentNode;
                    delete_button_change('active');
                }else{
                    console.log(number + " is selected.");
                    var el = document.querySelector(".line_selected");
                    if(el!=null)el.className = "line";
                    line_area.className = "line_selected";
                    tool_bar_switch('assign');
                    delete_button_change('active');
                }
            }
            console.log(line_area);
            if(line_area.parentNode.className=="box"){
                var if_div = line_area.parentNode.querySelector(".if_insert");
                if_div.innerHTML = "Export";
            }
            selected_line_num = number;
        }
    }
}

var insert_line = function(serial,line_div){
    var val = search_line(selected_line_num);
    var if_line = search_line(serial);
    var l = if_line.inner_lines.length;

    search_line(selected_line_num,0,lines,null,"Delete");

```

```

        if(line_div.querySelector(".if_insert").innerHTML=="Insert"){
            if_line.inner_lines[l] = val;
        }else{
            lines[lines.length] = val;
        }
        selected_line_num = -1;

        add_all_lineView(add_div,lines);
    }

    var if_option_view = function(serial,if_div,flag){
        if(selected_line_num!=-1&&selected_line_num!=serial){
            var ex = if_div.querySelectorAll(".if_option");
            var ex0 = if_div.querySelector(".if_rest");
            var res;
            var res0;
            if(flag){
                res = "block";
                res0 = "none";
            }else{
                res = "none";
                res0 = "block";
            }
            for(var i=0;i<ex.length;i++){
                ex[i].style.display = res;
            }
            if(ex0)
                ex0.style.display = res0;
        }
    }

    var line_change = function(n,n0){
        //      selected_lineと選択されたlineを入れ替える
        console.log(n+"<=>" +n0);
        var nLine = search_line(n,0);
        var n0Line = search_line(n0,0);

        if(check_child(nLine,n0Line))exchange_line(n,n0);

        add_all_lineView(add_div,lines);
    }

    var check_child = function(n,n0){
        //どちらかがブロックで、もう片方を内包していないかをチェックする
        /*
        1 if(){
            2 alert();
        }
        これで1と2を交換するとすごいことになるので避ける
        */
        var check = function(a,b){
            if(a.inner_lines){
                var ln = a.inner_lines;
                for(var prop in ln){
                    console.log(b.number+" ## "+ln[prop].number);
                    if(ln[prop].number==b.number){

```

```

        window.alert("ERROR!");
        return false;
    }
}
}
return true;
}

if(!lcheck(n,n0)||!lcheck(n0,n))return false;

return true;
}

var tb_new_var = function(){
    var nv = document.getElementById("new_var");
    var txt = nv.querySelector(".tool_text");
    var ipt = nv.querySelector(".tool_input");

    if(txt.style.display=="none"){
        txt.style.display = "block";
        ipt.style.display = "none";
    }else {
        ipt.querySelector("input").value = "";
        txt.style.display = "none";
        ipt.style.display = "block";
        ipt.querySelector(".text").focus();
    }
}

var tb_new_assign = function(){
    var hoge = document.getElementById("assign_new_var");
    var txt = hoge.querySelector(".tool_text");
    var selector = hoge.querySelector(".assign_selector");

    if(txt.style.display=="none"){
        txt.style.display = "block";
        selector.style.display = "none";
    }else {
        txt.style.display = "none";
        selector.style.display = "block";
    }
}

var tb_fixed_num = function(){
    var hoge = document.getElementById("assign_num");
    var txt = hoge.querySelector(".tool_text");
    var input = hoge.querySelector(".tool_input");

    if(txt.style.display=="none"){
        txt.style.display = "block";
        input.style.display = "none";
    }else {
        txt.style.display = "none";
        input.style.display = "block";
    }
}

```

```

var tb_fixed_str = function(){
    var hoge = document.getElementById("assign_str");
    var txt = hoge.querySelector(".tool_text");
    var input = hoge.querySelector(".tool_input");

    if(txt.style.display=="none"){
        txt.style.display = "block";
        input.style.display = "none";
    }else {
        txt.style.display = "none";
        input.style.display = "block";
    }
}

var tb_new_if = function(){
    var hoge = document.getElementById("new_if");
    var txt = hoge.querySelector(".tool_text");
    var input = hoge.querySelector(".tool_input");

    if(txt.style.display=="none"){
        txt.style.display = "block";
        input.style.display = "none";
    }else {
        txt.style.display = "none";
        input.style.display = "block";
    }
}

var tb_new_while = function(){
    var hoge = document.getElementById("new_while");
    var txt = hoge.querySelector(".tool_text");
    var input = hoge.querySelector(".tool_input");

    if(txt.style.display=="none"){
        txt.style.display = "block";
        input.style.display = "none";
    }else {
        txt.style.display = "none";
        input.style.display = "block";
    }
}

var make_new_if = function(){
    add_new_line("If",null,null);
}

var make_new_while = function(){
    add_new_line("While",null,null);
}

var make_new_assign_line = function(){
    var div = document.getElementById("assign_new_var");
    var selector = div.querySelector("select");
    var left_var = selector.value;
    if(left_var!="NO_VARIABLES"){
        add_new_line("Assign",left_var,"BEFORE_ASSIGN");
        tb_new_assign();
    }
}

```

```

    }
}

var remake_assign_line = function(value,type){
    var l = selected_line_num;
    if(l!=-1){
        var line = search_line(l);
        var par = search_line(l,0,lines,null,"ParentBox");

        if(line.kind=="Var"){
            var name = line.name;
            add_new_line("Assign",name,"BEOFRE_ASSIGN");
            l = par.length-1;
        }
        var selected_line = par[l];

        par[l].struct = new struct(value);
        par[l].assign_value = value;
        par[l].type = type;
        selected_line_num = -1;
        delete_button_change('stop');
        add_all_lineView(add_div,lines);
    }else{
        window.alert("Please Select Line...");
    }
}

var submitStop = function(e){
    if(!e){
        var e = window.event;
    }
    if(e.keyCode==13){
        return false;
    }
}

var tb_make_new_var = function(e){
    if(!e)
        var e = window.event;
    if(e.keyCode==13){
        new_var_hoge();
    }
}

var new_var_hoge = function(){
    var tbNewVar = document.getElementById("new_var");
    add_new_line("Var",tbNewVar.querySelector("input").value,null);
    tb_new_var();
}

var tb_assign_fixed_num = function(e){
    if(!e)
        var e = window.event;
    if(e.keyCode==13){
        assign_fixed_hoge();
    }
}

```

```

var assign_fixed_hoge = function(){
    var div = document.getElementById("assign_num");
    var num = div.querySelector("input").value;
    if(isNaN(num)){
        window.alert(num+" is not Number...");
    }else{
        num = parseFloat(num);
        remake_assign_line(num,"Number");
        div.querySelector("input").value = "";
        tb_fixed_num();
        tool_bar_switch('new');
    }
}

var tb_assign_fixed_string = function(e){
    if(!e)
        var e = window.event;
    if(e.keyCode==13){
        assign_fixed_string();
    }
}

var assign_fixed_string = function(){
    var div = document.getElementById("assign_str");
    var str = div.querySelector("input").value;

    remake_assign_line(str,"String");
    div.querySelector("input").value = "";
    tb_fixed_str();
    tool_bar_switch('new');
}

var delete_button_change = function(hoge){
    var dButton = document.getElementById("delete_button");
    dButton.className = hoge;
}

var delete_line = function(){
    if(document.getElementById("delete_button").className=='active'){
        search_line(selected_line_num,0,lines,null,"Delete");

        selected_line_num = -1;
        add_all_lineView(add_div,lines);
        document.getElementById("delete_button").className = 'stop';
    }
}

var make_code = function(){

    var code_roop = function(layer,box){
        var txt = "";
        for(var prop in box){
            var l = box[prop];
            console.log(layer);
            for(var i=0;i<layer;i++)
                txt += "  ";
        }
    }
}

```

```

        if(l.kind=="Assign"){
            txt += l.name + "=" + l.assign_value + "<br>";
        }else if(l.kind=="If"){
            txt += "if"+l.condition+"<br>" + code_roop(layer+1,l.inner_lines) + "<br>";
        }else if(l.kind=="While"){
            txt += "while"+l.condition+"<br>"+code_roop(layer+1,l.inner_lines) + "<br>";
        }
    }
    return txt;
}

console.log(lines);
var code = "";

for(var prop in variables){
    code += ("var " + variables[prop].name + "<br>");
}

code += code_roop(0,lines);

var storage = sessionStorage;
storage.setItem('code',code);

var x = window.open("script.html","");
}

var put_code = function(){
    var storage = sessionStorage;
    var code = storage.getItem('code');
    document.getElementById("code_area").innerHTML = code;
}

var addIf = function(num){
    if(lines[num].kind=="If"){
        var div = document.querySelector(".box_selected");
        add_new_lineView(1,"NEW_VARIABLE","a");
        div.className = "box";
        return "If";
    }
}

var search_line = function(serial,layer,list,assign_line,type){
    // serial_numberでlineを検索する
    if(layer==0||!list){
        list = lines;
    }

    for(var prop in list){
        if(list[prop].number==serial){
            if(assign_line){
                console.log(assign_line.name+"==>" +list[prop].name);
                list[prop] = assign_line;
                return true;
            }else{
                console.log("==>" +list[prop].number);
                answer = list[prop];
                if(type=="Delete"){

```



```

        list = list.splice(prop,1);
        console.log(list);
        return true;
    }else if(type=="ParentBox"){
        return list;
    }else{
        return list[prop];
    }
}
}else if(list[prop].kind=="If"||list[prop].kind=="While"){
    console.log(list[prop].kind + " BLOCK");
    var res = search_line(serial,layer+1,list[prop].inner_lines,assign_line,type);
    if(res)
        return res;
    if(list[prop].else_array&&list[prop].else_array.length!=1){
        var res = search_line(serial,layer+1,list[prop].else_array,assign_line,type);
        if(res){
            return res;
        }
    }
}
}

if(layer==0&&!answer){
    console.log(serial+" is NOT FOUND...");
}

}

var exchange_line = function(n1,n2,layer,list){
    if(!list||!layer){
        list = lines;
        layer = 0;
    }

    var add,res,pl;

    for(var prop in list){
        if(list[prop].number==n1){
            res = n2;
            if(layer!=0){
                return res;
            }
        }else if(list[prop].number==n2){
            res = n1;
            if(layer!=0){
                return res;
            }
        }else if(list[prop].kind=="If"||list[prop].kind=="While"){
            res = exchange_line(n1,n2,layer+1,list[prop].inner_lines);
        }
        if(res&&layer==0){
            if(res==n1)add = n2;
            else add = n1;
            pl = search_line(res);
            search_line(res,0,lines,search_line(add));
            break;
        }
    }
}

```

```

    }

    if(layer==0){
        search_line(add,0,lines,p);
    }
}

window.onload = function(){
    project = document.getElementById("project");
    window.addEventListener("mouseup", function(){window_drop()}, false);
    var t = document.getElementById("tb");
    lines[0] = new line(0,"Start");
    selected_line_num = -1;
    selected_struct_num = -1;
    serial_number = 1;
    variables = new Object();
    add_div = document.querySelector("#struct_area");

    var as0 = document.getElementById("var_list_bar");
    var co0 = document.getElementById("other_list_bar");
    as0.addEventListener("click", function(){tpChange0("varList")},false);
    co0.addEventListener("click", function(){tpChange0("otherList")},false);

    var as = document.getElementById("type_assign");
    as.addEventListener("click", function(){type_change("Assign")},false);
    var co = document.getElementById("type_condition");
    co.addEventListener("click", function(){type_change("Condition")},false);

    var other_num = document.getElementById("other_num");
    other_num.addEventListener("mousedown", function(){num_drag_start()}, false);

    var other_str = document.getElementById("other_str");
    other_str.addEventListener("mousedown", function(){str_drag_start()}, false);

    var target = document.getElementById("target");
    target.addEventListener("mouseup",function(){drop(target)},false);

    var insert = document.querySelector(".insert");
    insert.addEventListener("click", function(){assign_export_window()},false);

    var restart = document.querySelector(".restart");
    restart.addEventListener("click", function(){assign_restart_window()}, false);

    var trash = document.querySelector(".trash");
    trash.addEventListener("mouseup", function(){trash_drop()},false);

    var condInsert = document.querySelector(".cond_insert");
    condInsert.addEventListener("click", cond_insert_window, false);

    var condRestart = document.querySelector(".cond_restart");
    condRestart.addEventListener("click", cond_restart_window,false);

    var condTrash = document.querySelector(".cond_trash");
    condTrash.addEventListener("mouseup", function(){trash_drop()},false);

    add_null();

```

```
        add_addButton();  
        vList_addButton();  
  
        make_cBox();  
        make_cAdd();  
  
        console.log(add_div);  
    }  
}
```

A.4 script.html

```
<!DOCTYPE html>
  <head>
    <meta charset="UTF-8">
    <title>code page</title>
    <script type="text/javascript" src="code.js"></script>
  </head>

  <body onload="put_code();">
    <div id="code_area"></div>
  </body>
</html>
```