この文書は20ページあります

易しくない パソコン リテラシー(1/4)
ソフトウエア編

0. あらまし

この文書は、「**パソコンを社会人並みに考えて、その人に仕事をしてもらう実用文書の書き方**」の解説です。コンピュータプログラミングは、英作文の知識で作成する応用技術です。英語の方言のような文体で書くことになりますので、「**易しくない**」ですよ、と表題の頭に付けました。文書全体は、A4版 80 ページあります。20 ページにまとめた 4 分冊に編集しました。主題は第4編の「コンピュータプログラミングの演習」教材です。パソコンは、現代の科学技術が応用されている最先端の道具です。「技術は三つの要素から構成されている」と定義することができます。「道具・技法・技能」です。現代風に言えば、「ハードウエア・ソフトウエア・ユーザインタフェース」です。演習の場合、パソコンその物についての基礎的な知識が必要ですので、3つの要素名を見出しにした3分冊を前半に構成してあります。専門用語の知識が無いと、全く理解ができません。この文書は、結果的には用語解説を主な目的とした編集になりました。

小学校低学年の基礎教育の段階では、「読み・書き・数え」が教えられます。慣用は、「読み・書き・ソロバン」です。パソコンの利用が普及してきましたので、社会人の教養として「読み・書き・パソコン」が話題になってきました。これをぴったりと表す英語が Computer Literacy です。リテラシーは、カタカナ語で使われていて、「読み・書きの素養」のことです。学問用語には向きませんので、大学などでの学科名に使われることはありません。義務教育の場面に加えることの是非が、話題になってきました。しかし、幾らか無理があります。それは、英語の知識が必須だからです。大学の教養科目とするのが良いでしょう。欧米の大学では、言語違いで多くの学生が入学してきますので、従来から、Technical Writing (実用文書の作成)の教科がありました。ところが、「読み・書き・ソロバン」に続けるべき、重要な基礎教養である、「聞く・話す」の対話(communication)が抜けていることは、問題にされてきませんでした。技術教育では、実際に経験する場が必要です。参考書を自習してお勉強をすることも大切ですが、経験のある人との対話をで手ほどきをしてもらうと、簡単に納得できることが多いものです。この文書は、対面授業(interface; 顔合わせ)が大きな効果を上げる教育方法であることの説明資料です。また、「**です・ます調**」の文体で書きました。声に出して相手に話し掛けるとき、丁寧な言い方になることを意識したためです。また、自分で発声し、それを自分の耳で聴くことが、非常に有用であることの納得が得られることも、意識して編集しました。

2022-7: 島田 静雄

目 次

用 語 索 引

現代はインターネットで検索すれば、多くの情報が得られます。しかし、検索に使う用語を知っていなければ利用できません。この文書の中では、インターネットで検索すれば、詳しい説明が得られる用語は載せて無い場合があります。インターネットでの情報では、説明に満足できないこともあります。索引は、文書の編集では全体の終わりに付けることが習慣ですが、筆者の文書では頭の項目にしてあります。番号はページ番号ではなく、章・節・項の番号です。目次も索引的な使い方をすることにして、目次に出てくる用語は、下の索引には載せてありません。

1. 雑学的言語学(ソフトウエア編)

1.1 言葉を学問対象として扱う

1.1.1 文字で表した言語で研究する

言語学は、linguistics の訳であって、基本的には、**音声**を研究する**学問**(study)です。音声は、一過性の事象です。テープレコーダのような、記録と再生のできる装置が無かった時代、記録が残っている文字並びで**言語**(language)を**研究**するしか方法がありませんでした。現代でも、文字並びで言語の研究をすることが主流です。日本は、中国大陸から見れば孤立した島国であって、固有の言語が使われています。この言語のルーツがどこに在ったかは、よく分かっていません。南アジア圏から、島伝いに海洋民族が渡来してきて使うようになった言語のようです。言語構造が中国語では SVO(主語・動詞・目的語)の順であるのに対して、日本語では SOV の順になっている違いがあるからです。中国大陸と文化的な交流が始まったのは、歴史的には西暦前10世紀頃の弥生時代からのようです。しかし、言語構造を変化させることは無かったようです。文字の輸入と、その利用が盛んになったのは7世紀頃からとされています。**話し言葉**は、世代が代わっても、また、地域ごとの方言違いがあっても、大きな変化をしません。しかし、文字の利用は、大きな変化をもたらします。江戸時代には、断片的にしか西欧事情が知られていませんでした。明治維新(1868)以降、欧米文化を積極的に学ぶことが勧められました。これは、漢字の輸入に次ぐ第二波の学びの時代です。日本の敗戦後(1945)は、第三波の学び時代の始まりです。学問の研究には、主に欧米文書が使われましたので、欧米言語を学ぶことが勧められてきました。

1.1.2 学問は過去の事象を扱う

欧米文化を学ぶことは、既に欧米で知られている**事実**(known facts)や**法則**(theory)などを、**知識**として取り込む方法を採ります。これは、視線としては過去を向いています。**学**(まな)**ぶ**とは、**真似**(まねぶ)の転化した語です。学びの対象は、(…学)と名付けます。戦後、専門を細かく分類して、(…学)として分化することが増え、「自分こそが(…学)の本家である」のような権威付けをする風潮があります。一方、産業と関係のある**技術**(technology)は、手っ取り速く、製品を輸入することに始まります。自前で研究・開発をすることは、視線が未来を向きます。何が起こるかの予測ができなくて、賭け事の様な面があり、危険を伴うこともあります。知識の多さにこだわる科学者は、手を出さないことがあります。これは**保守的**な態度です。技術は、**進歩的**な考え方を必要としますので、対立も起こります。技術は、実用されることで評価されます。実用主義を(pragmatism)**プラグマティズム**と言い、アメリカでは根本的な思想です。コンピュータ関連の技術が、アメリカ主導型で開発・研究されてきたことの背景にはプラグマティズムがあります。ヨーロッパでの権威主義的な観念論(**イデアリズム**)とは、一線を画します。また中国起源の朱子学(**儒教**)とも対立します。明治維新は、この三つの思想が複雑に絡み合う時代の始まりでした。文化的にはヨーロッパに多くを学びました。アメリカからは、実用技術を多く学びました。「少年よ大志を抱け」を言ったクラーク(1826-1886)が招聘された札幌農学校がそうでした。徳川幕府に反抗していた水戸藩は、朱子学を主導していましたので、明治新政府は王政復古・尊王攘夷も旗印にした神仏分離令を発布して仏教を標的にしました。中国の文化大革命と似た面があります。そのため、かなりの寺院が被害を受け、多くの文化財が消失しました。

1.1.3 研究は分類から始める

科学(science)は、全体を細かく分類し、研究したい部分だけを取り出す方法の意味があります。**抽象**と言います。「象(しょう)」は、和語では(かたち)と読みます。同じ意義で**捨象**があります。余分なものを捨てることを言います。抽象の過程は、全体を分類して、対象を選り分ける方法を採ります。リンネ(1707-1778)が始めたとされる生物分類法は、広く応用されるようになりました。全体を見渡す方法の一つが、グラフ理論のツリー構造です。階層的な分類に使う漢字の一般的な接尾辞は「界・門・項・目・科・属・種」です。科学は、科の字が使われている用語です。その研究方法は、過去の**データ**(資料)を集め、そこには「法則性が有るはずだ」との思い込みがあります。分類作業では、**統計**的な解析方法も応用されます。一方、技術は、「これから何かを作る」のように、視線が未来を向きます。何が起きるかの予測ができないこともあります。例えば、橋は社会環境では必要になります。技術的には物騒な構造であって、落橋などの事故の記録が多く残されています。安全な完成を神に願う、などの行為は、科学的な態度ではないのでしょうが、現代でも行なわれています。したがって、架橋の意思決定は、賭け事のような扱いがあります。潔癖な科学者は、**確率**的な解析手法での未来予測を嫌います。アインシュタインは、「神はサイコロを振り給わず」の言葉を残しました。何かの自然事象は、必ず法則性がある、との科学万能信仰からです。予測のできないことの研究、例えば地震予知の研究は、法則が未だ発見されていないからだ、と言う立場をとります。しかし、確率・統計的な方法は、全体から細部に見方を狭めていく考え方の一つです。一般的ですので、科学者も使う実践的な方法です。

1.2 言葉を覚え・使う過程

1.2.1 言語が使われる環境

環境(environment)は、よく使われるようになった用語です。**自然環境**と**社会環境**の区別があります。広く繋がった社会環境は、一つの**言語圏**を構成します。人が多く集まる都市の規模は、戦前までは、1辺 8 ㎞(2里)の正方形、または半径 4 ㎞(1里)の円の範囲程度でした。その日の中に歩いて往復できる**社会環境**でもありました。明治時代前、江戸・京都などでも、それよりやや広い程度の規模でした。世界的にみても、常識的な広さでした。路面電車の利用は、都市の規模を約 2 倍にする効果がありました。自家用車時代になって、都市規模は巨大化に進んでいます。都市は、主に**第二次産業**(工業など)、**第三次産業**(商業・学芸など)の場であって、商人・職人、そして学習者などが集まります。都市の外側は、**第一次産業**の場であって、材料・食料などの生活物資の供給元です。都市部の社会環境は、人の交流が多くなりますので、**文字**の利用と標準的な**言葉遣い**が育ちます。周辺では方言が残ることが見られ、**田舎っぺ**と馬鹿にする軽蔑用語の扱いがあります。江戸時代、地方の大名は参勤交代が義務付けられていましたので、結果的に方言違いを均す標準的な江戸言葉が普及しました。明治になって、北海道へも多くの入植者が移りました。ここでも、江戸言葉に近い標準化した言語が育ちました。幾らか違いもあって、東京では**鼻濁音**が強く使われています。近代になって、通信手段が進歩してきたことで、直接の顔合わせ(インタフェース)をしない社会環境が生まれてきました。単純に考えると、近代化が進んできたと好意的に受け取られていますが、引き篭もりや孤独化による負の面が問題になってきました。

1.2.2 読み・書き・数え・聞く・話す

そもそも、人が生を得た赤ちゃんのときから、幼児に成長するまでの短期間に、大人(おとな)が使っている話し言葉を覚える過程は、神秘的です。母親が話しかけ、それを聞いて動物的に反応し、発声で母親に伝える方法で覚えます。これが**母語**(ぼご:mother tongue, mother language」です。家族関係が濃い集団を**民族**と言い、そこで話す言葉を native language と言い換え、**母国語**(ぼこくご)と当てています。母語違いは、**方言**(dialect)違いができる原因の一つです。小学校の段階では、「**読み・書き・数え**」の義務教育が組まれます。ところが、言語としての基本である「**聞く・話す**」の教育は、明示的には扱われていません。戦後、英会話の勉強がブームになりました。しかし、英語の**ネイティブスピーカー**が身近に居ないため、実践的な対話の機会が殆どありませんでした。大学入試の前段階に、いわゆる共通試験があって、英語の**聞き取り**試験が組み込まれるようになりました。しかし、**話し方**の方の試験は、海外と交流のある企業の入社試験の場で行なわれることもある、と言うのが現実です。音声による対話には、丁寧な**言い方**が教養として必要です。小学校の低学年までは、近所では、うるさいと嫌われるほど、声を出させる教育をしています。引き籠りや不登校は、声の無い閉鎖的な環境になっている問題と考えることができます。また、丁寧な言い方ができなくて、結果的に、「無礼・舌足らず・いじめ的(**ハラスメント**:harassment)な言葉遣い」になることが起こります。戦後の日本国憲法では、表現の自由が権利として認められるようになりました。しかし、発話に関して言えば、発言の**自由**と**わがまま**との境界が明確ではないことと、守るべき倫理的な義務に無頓着になることが社会問題化するようになっています。

1.2.3 パソコンが使う母語

パソコン(personal computer)が普及してきたことで、パソコン内部で使う言語(バイト)を工夫し、対話(**インタフエース**)を考えるようになりました。パソコン内部で交錯する電気信号のコードが、**バイト**(byte;)の源形です。「binary term」から作られた**造語**です。バイトは、パソコンを擬人化し、その人が使う**母語**を構成する最小単位です。パソコン内外部の送受信コードに使います。複数の信号線(6 ないし 8)で装置間をつないで利用する(**並列インタフェース**)と、一本の信号線を使い、時間的に(On・Off)のパルス状の信号を使う(**直列インタフェース**)とがあります。後者には、俗に「トン・ツー」と言うモールス信号での**有線**通信があり、日本海海戦(**1905**)の時は、**無線**通信で利用されました。現在でも、国際的なアマチュア無線で使われているコード系です。バイトは、最小の語単位です。眼に見えない情報ですので、法律的には**無体物**です。送受信される情報は、**財物**扱いをするようになり、ソフトウエアと総称し、広範囲の**情報産業**を構成するようになりました。同じラテン文字を使っても、英・独・仏の言語は、語の種類と並べ方に区別があるように、複数のバイト並びを、**ワード**と言う単位で使うことがあります。語・句の集合で**文**を構成します。文の集合が**文書**です。バイト単位の段階では、ソフトウエア的には、三通りの使い方があります。第一は**数値**、第二は**文字**です。数値と数字とでは、ビット並びは別定義です。第三は、プロセッサに指示する**命令コード**(オペレーションコード)です。普通の電卓では、加減乗除のキー(＋－×÷)を押すと、命令コードがレジスタに送信されます。パソコンの場合には、**目的語**を従えた**他動詞**です。その集合で、或るまとまった変数・関数・処理を組み立て、名前(キーワード)を付けます。プログラマが命名してパソコン側に知らせます。その名前のことを**識別子**(identifier)と言います。原則として、頭字を英字で始める英数字並びです。パソコン側で決めてある**予約語**(または**キーワード**)も使わない約束です。

1.2.4 音声による伝承がある

人類が文字を利用し始めたのは古代エジプト、紀元前 3200 年頃から、アルファベットの原点となった**ヒエログリフ**から、とされています。石版のほか、紙(パピルス)の利用もありました。**ハムラビ法典**は、楔型文字が紀元前 3400 年頃に使われ、中東のメソポタミア文明を支える統治に利用されていたようです。日本の縄文時代は、紀元前一万年以上も前から続いていたとされています。文字の利用は無かった時代です。音声は一過性の事象です。何も記録が残りません。ところが、神話や伝承は、何代にも亘って口伝えで伝えられた例があります。日本では、アイヌ民族に伝わる叙事詩、ユーカラが知られています。変質しないで、かなり正確に伝えられてきました。古事記や日本書紀は、それまでの伝承を、当時の最新文化の漢字を使って固定した、と考えることができます。内容に怪しい記述もありますが、必ずしもフィクション(虚構)とは言えない史実も記録された、と想像しています。世界的に見ても、このような伝承記録があって、例えば、旧約聖書にある**ノアの方舟**(はこぶね)、ギリシャ神話の**トロイの木馬**の記述は、当時、良質の森林資源が有ったことを暗示しています。文字による記録が無かったため、歴史が良く分からない地域があります。例えば、アフリカ南部は、文字が使われなかったこともあって、歴史が分からず、欧米人は暗黒大陸と蔑称しました。

1.2.5 音声の文字化

人が話す言葉は、物理的には音(sound)です。情報を伝える意思を含めるときが**話す**です。音声は**一過性**の事象であって、何も残りません。そこで、音声を記録する方法に図形を使います。**字**または**文字**と言うのが普通ですが、**絵文字**もあります。最小の音声単位を**音素**と言います。英語のアルファベットは、母音(ぼいん)と子音(しいん)を持つ 26 個の**表音文字**(letter)を使います。日本語の仮名は、50 音と約2倍あります。ただし、濁音と半濁音が 25 音別にあります。母音と子音の発声の対を表すためです。アルファベットと仮名とは、表音文字です。音素並びの最小構成を**音節**(syllable)と言い、複数の音節並びが**語**(word)です。意味を持たせる最小単位です。この種類を言うとき**品詞**(parts of speech)に分けます。日本語での品詞の考え方は、明治以降、英語辞書の編集に倣って、**大槻文彦**の「**言海**」で採用されたことが初めです。語並びもアイウエオ順です。当時はいろは順が普通でしたので、画期的なことでした。したがって、いろは順の索引も付いています。漢字を品詞に分類した漢和辞典は、未だありません。中国では一字一音節で使います。語としての意義を持ち、数千種類の**字形**があります。漢字は**表意文字**(character)である、と言います。**形態素解析**は、プログラミング言語の文字列解読に使うコンパイラに応用されるようになった、1990 年頃から再研究がされはじめた品詞論です。

1.2.6 日本は個性的な一言語社会である

陸続きのアジア・ヨーロッパの大陸では、種々の言語が使われています。日本は、孤立した島国です。方言違いはありますが、一言語が使われています。**識字率**が高いため、口頭での対話が少なくても済みます。日本語の書き言葉は、**漢字**と**仮名**(かな)とを混用しています。その漢字も、中国語を元にした**音読み**と、和語としての**訓読み**もあって標準化・一元化はできていません。明治維新後、日本は、欧米文化に多くを学びました。漢字は、字種が数千もあって覚えることが難しいため、**表音文字**のアルファベットを採用したいとする提案もありました。読みと書きとの共通化の提案に**言文一致**運動が起こり、これが今日の口語文スタイルへと進んできました。そうであっても、「**です・ます調**」と「**である調**」の二種のスタイル違いが続いています。この文書では、筆者は、意識して「です・ます調」の文体で書きました。できるだけ和語の言い方を採用しています。漢字を比較的多く使いますが、難しい読みは括弧に(かな読み)を添えるようにしています。ワープロでは小文字を使う振り仮名表記ができますが、HTML 文書には利用できませんので、使いません。同じ理由で、少し読み難くなるのですが、段組みも使いません。

1.2.7 宗教に文字が利用される

政治的な統治に宗教が利用されることも、多くみられます。宗教は、排他的(exclusive)で攻撃的(offensive)な集団を構成することが多く、その拠り所に教典のような文書をまとめて、結束が図られます。ユダヤ教の旧約聖書、キリスト教の新約聖書、イスラム教の聖典、仏教では仏典(三蔵)です。殺戮・破壊・略奪を、何がしかの正義を掲げて正当化することもします。江戸幕府がキリスト教を警戒したことの理由は、その攻撃性にありました。大陸経由で伝来した仏教も、攻撃的な面を持っていました。山城を構え、僧兵が攻撃と防備に当たりました。弁慶は薙刀を持ち、比叡山延暦寺の僧兵でした。仏教が攻撃的な面を持っていたため、武家集団との対立も起きました。仏教も、山城から平野にある居住地域に降りてきましたが、塀で囲う閉鎖的な構築です。「…主義」の用語は、近代になって目につくようになりましたが、宗教の現代版の性格が有る、と見れば納得できる面があります。これも攻撃的な信条を文書化していて、「闘争」の用語が随所に使われています。社会主義・共産主義は、日本のインテリ層に受ける面がありますが、武装闘争は自然災害の多い日本風土にはなじみません。

1.2.8 日本の宗教観

日本では、**神道**(しんとう)で代表されるように、自然崇拝が基本にあって、大らかです。欧米から見るとき、**アニミズム**(animsm)であると軽蔑的に言うことがあります。しかし、日本の国土は自然災害が多いため、自然は怖い面もあり、自然と**共存する**考え方が必要です。神は、何かの専門を持っている霊魂と考えています。生活環境の安全を祈願する対応が見られます。仁徳天皇の前方後円墳が残っているように、立ち入り禁止の聖域を考えることは、安全対策になっていると考えることができます。欧米の宗教は、日本人から見ると人間的な生臭さが感じられます。日本の宗教観には、幾つかの特徴がありますが、概して開放的です。相撲(角力)が神道の行事に組み込まれる場面では、形式美があります。その場合には、土俵は聖域扱いをします。女性が上がることはタブーですが、安全対策になっています。**ご神体**は神が宿る物体を言い、それを祀り、礼拝の対象です。神は、普段、山に居るとされています。したがって、山は基本的にはご神体であり、また、聖地です。富士山は、ご神体です。したがって、山を「征服する」のような言い方をしません。イスラム教と同じで、偶像崇拝もしません。ただし、絵画や彫刻での芸術的な表し方はします。これは、文化の継承には非常に重要な態度です。何かの願い事をするとき、神主(かんぬし)が祈祷して身近の祭壇などに呼び出します。願い事に使う言語が**祝詞**(のりと)です。願い事が済めば、お帰りをお願いします。後は酒盛りです。悪霊もあって、出てきては迷惑であるとして、閉じ込めることを目的とした神社もあります。俗信と見なされていますが、旧暦の 10 月を神無月(かんなづき)と言い、全国の神が出雲(いずも:島根県)に集まり、言わば年次総会(annual meeting)を持つため不在になる月と考えています。出雲では、10 月を、逆に神在月(かみありづき)と言うのだそうです。

1.2.9 書く と話す とが正確に覚える方法

人の眼は二つあります。全体視野で、物を立体的に理解しています。しかし、**注目**をしている範囲は非常に狭いことに気付いているでしょうか？　作文をして、その校正作業をするとき、一字一字を注意深く**注視**して確かめることをしないと、誤字・脱字のミスを見逃します。モニタ上での校正作業と共に、校正刷りを作ると、見逃していたミスを見つけることがあります。交通事故を起こした人が、「前をよく見ていなかった」の釈明をみます。注目していた場所が違っていたからです。明視の距離で読む文書の寸法は、A6 版(ハガキ大)から A4 版てす。文書は、視線を合わせている字は読めますが、それを少し外れた箇所の字は読めません。横長に並べた文字列を読むとき、左から右へと視線を移動し、行を換えで左端に戻すとき、行を間違えることがあります。一行幅を狭くする**段組み**は、読み易さを助ける物理的な方法です。同じように見える二つの絵を見比べて、違いを探すクイズがあります(図 1.1)。左・右・左…と視線を変えて注視をしなければ見つけられないのです。

図 1.1　間違い探しのクイズ(中日新聞カレンダ 2022-4)

日常生活の場では、音声を介する対話が主であって、文字を媒介とした交流をしなくても済みます。ただし、漢字圏では、筆談を使う方法があります。これが表意文字である漢字の利点です。文字が理解できないことを**文盲**(もんもう)と言い、差別用語とされていますので、**識字**(しきじ)の用語の方を使います。「文字の書き」と「声に出す話し」とでは、微妙な違いがあります。文字並びは、声に出し、それを自分の耳で聴くフィードバック(**再帰**)を掛けていることで理解しています。知識の吸収には高い効果があります。電車やバスの運転士が、**指差し呼称**で信号確認に声を出すことは、安全運転に非常に重要です。**黙読**をしているときでも頭の中では音で理解しています。子供は、黙読のときでも、口が動いています。読みが分からないところで、動きが止まります。漢字は表意文字ですので、大人は別読みや、読み飛ばしもしています。ところが、文字は声に出し、書かないと正しく覚えることができません。字をきれいに描くことは教養があるとして尊敬されます。**お習字**は、文字を覚えさせる教育に、想像以上の効果があります。英語の用語は calligraphy です。英語のように表音文字使う場合も、丁寧で、きれいに書く練習をすることで、スペル間違いをしなくなる、などの正確性が身に付きます。しかし、芸能人の落書きもどきのサインは、教養の無さを公開している恥さらしの面があり、お習字文化を破壊しています。

1.2.10 英語と日本語とはどこが異なるのか

英語の素養の必要性について補足しておきます。日本語は、英語・ドイツ語・フランス語とは、かなり異なります。この3つの欧米言語は、独立した別言語のように思われています。ところが、骨格構造(language structure, syntax)は、ほぼ同じです。**構造**(structure)の用語を使っていますが、正確に言えば、**英語風の文構造**(English-like-structure)と言う意味です。欧米人には常識であるため、English-like の語を省きますので、日本人が誤解します。言語は単語の並びです。単語は品詞で種類分けをします。**品詞論**として扱います。名詞・動詞・副詞・形容詞の四つが主な品詞名です。日本語・英語共に全部で 10 種類程度の区分です。品詞をどのような順に並べて言う、または書くかを論じるとき、**形態論**も扱われます。そこでは、品詞並びと共に、区切り符号を含めて、**形態素**と言い換えます。また、日本語の**助詞**「て・に・を・は」などは後置型です。英語での(in, on, at,…)は**前置詞**と言い、品詞名が別です。英単語は、人称・時制・単複で綴りが変る(屈折する)ため、**屈折語**と言います。区切り符号の使い方には規則がありますが、日本語には無い約束です。日本語では、品詞間に空白を使わない書き方をします。ペタリと繋げる文字並びですので、**膠着語**と言います。日本語では、品詞の区分が分からなくなることが起こります。よく引用される構文が「べんけいが、**なぎなた**を持って」が正しいのですが、「べんけいがな、**ぎなた**を持って」と誤読することを例に使います。因みに、中国語の構文は**孤立語**と言います。一字単位で独立させても意味を持たせることができるからです。

もう一つ、文型の違いがあります。英語と日本語とでは**語順**に違いがあることです。英語の文法書には、5種類の文型が紹介されています。基本は、英語では、第4文型 SVO と、第5文型 SVOO の順です、とりわけ、第5文型で前置詞を使わない(主語・動詞・間接目的語・直接目的語)の語順が基本です。日本語は SOV です。助詞の「に・を」後置詞に使う目的語では、前後の語順違いを神経質に区別しません。英語と日本語の構文違いを下にまとめます。

- 品詞の考え方は、新しい概念ですが、定義には揺れがあり、議論の多い課題です。英語では、単語を構成する文字並び間に区切りを入れます。代表的には、スペース・コンマ・ピリオドです。挿入方法に標準があります。日本語では、熟語間に区切りを入れる語並びの規則がありません。読みを助けるため、コンマ「、」や中点「・」などを恣意(しい)的(好きなように)に使います。
- 英語風のプログラム文は、**他動詞**先行の**命令文**で構成します。自動詞は、パソコンが勝手に動く意味になりますので、命令文はありません。他動詞的に使うとき、**使役動詞**の make, let と共に使います。「run」は、典型的な自動詞ですが、(execute)の意味であると載せてある英語のパソコン辞書があります。日本語の使役動詞は、例えば、「走らせる」と言います。FORTRAN 言語では動詞を使わない代入文を使います。GOTO 文も原則的には自動詞ですので、積極的な使い方を避けます。be 動詞は空気のような重要な自動詞です。省略もされます。英語の native speaker は文を誤解することはありません。さらに、副詞と形容詞も使いません。greater, less など数量の比較に使う語は、演算子に代える方が主流です。
- 英語では、名詞の単・複と冠詞の組み合わせによって、文意の区別をします。プログラム文では、日本語に訳すとき、舌足らずの表現か、誤訳になることがあります。ただし、コンピュータ言語で使う予約語の名詞は、単・複の区別をしないことと、冠詞も使いませんので、日本人の勉強には助かっています。
- 英語は、日常使う明確な**命令文**の構造があります。他動詞が最初にきます。日本語の動詞の活用形には、命令形はありますが、日常言語では、普通には「…して下さい」の形で文末に使います。プログラミングは、英語風に命令文の構造で作文するのが適しています。
- 疑問文に対する返事は、日本語では、「はい・いいえ」を使い分けます。相手の言うことが、自分に対して論理的に正しいときに「はい」と答えます。例えば、或るテレビ番組を「…見ましたか？」と「…見なかったのすか？」の二通りの問いかけを受けた時、自分が見ていなければ、前者の問いには「いいえ」と答え、後者の問いには「はい」と答えます。アジア系の言語は、この方式です。一方、英語の場合には、自分に当てはめた場合の動詞の肯定・否定に合わせて「yes・no」を使い分けます。
- 日本語では、通称で言う「れば・たら」の文構造があります。論理的な判断を言うときに使い、結果を「はい・いいえ」で区別します。英語では、**条件文**と言い、「if…then…else」の形です。複雑な条件が重なるときの分類法は、**二分木法**(binary tree)が応用されます。二つの区分のどちらでもない場面があります。「勝ち・負け」を決めたいとき、「引き分け」も使います。ジャンケンでは「あいこ」があります。便宜的に、この区別を、**三分木法**と言うことにしました。
- 三分木法は、FORTRAN77 の版には「**算術 if 文**」がありました。数の「マイナス・ゼロ・プラス」を判断して、どの処理を選ぶかを決めます。現在は使いません。論理の帰趨を区別するとき、日本語では「はい・いいえ」を使います。英語では「真偽」を区別するとき、「yes・no」で言うことができません。「true・false」を当てます。初級英語の教育では、true に代える「that's right」の言い方を教えます。

1.3 実用文の作成と編集

1.3.1 実用文書

社会環境では、文字を介した情報交換が必要になり、**学術論文**、**レポート**、**ビジネスレター**に代表される**実用文**の作成が行なわれます。詩歌や小説の作文とは区別します。私的にハガキや手紙を書くことも、実用文の作成です。絵画や写真などは**画像**と総称し、単独では実用文書になりません。工業図面は、産業活動での利用を目的とて作成する画像扱いをしますが、ここで言う実用文書(主)に添える(従)の使い方です。別に作成規則があります。義務教育の過程では「思ったことを書きなさい」と指導しています。相手に分かり易く説明するなどの目的意識が無いことがあります。実用文書は、意味を伝えるための**論理的**な構成と、見てくれと関係する**物理的**な構造と、それぞれに、三つの要素を考えます。

- 論理的な構成は「**頭の項目**(front matter)・**本体**(body)・**終りの項目**(end matter)」の三部構成です。明示的な説明をあまり見ませんが、非常に重要です。三部分それぞれに独立したページ番号、数字も別字形を当てることがあります。因みに、三つの構成要素で作文する文芸に**俳句**があります。話芸に**落語**があります。「まくら・本題・落ち(サゲ)」で構成します。大衆演芸に**講談**もあります。これらは、日常の話し方を覚える場として、外国人にも評価されています。
- 物理的な構造は、「**書式**(format)・**様式**;style)・**文**(sentence)」です。**段組み**は、物理的な書式の一つです。語感が似ていますが、**段落**(paragraph)は論理的な文章のまとまりを言うときの単位であって、その区切りは改行です。日本語の作文教育で抜けている項目が、書式の**段落**構成法です。

1.3.2 語・句・文・文書

文書構成の最小単位が**字**です。**語**(word)は、一つ以上の字並びで、意味を表すときの最小構成です。**句**(phrase)は、**語句**、**熟語**、**熟字**とも言います。辞書で語の説明をするとき、文法的には、複数の**品詞**(parts of speech)に分類します。意味を持たせるように構成した語並び単位が、**文**(sentence)です。英語では、一単位の文は、英大文字で始め、数字(digit)、記号(symbol)なども混じり、ピリオド(period または full stop)までです。一文の文字並びの中で、品詞がどのような機能を区別するかの用語が、主語(subject)、述語(verb)、目的語(object)です。その並びの規則が、通称で言う、英語では **SVO**, 日本語では **SOV** です。複数の文の集合で、論理的な主張を**段落**(パラグラフ;paragraph)にまとめ、復帰(carriage return)と改行(line feed)とで明示的に区切ります。必要に応じて、段落に**見出し**(ラベル;label)または**表題**(title)をつけ、複数の段落の集合で**文書**(document)単位の作文としてまとめます。

一般社会の場では、実用文書の書き方は、弁えるべき基礎教養の一つです。書店に行くと、実用書の分類棚に参考書が並びます。官公庁を含め、一般企業でも作成する文書は**公用文**と総称し、代表的な実用文書です。その書き方については規格があります。参考にするべき、文献資料を、三つ上げます;

- **JIS Z 8301-2019**:規格票の様式及び作成方法、日本規格協会
- **公用文作成の要領**;文化庁(平成 5 年、用紙寸法に A4 版を採用するように改正されました)
- **ISO TC46:** Information and Documentation:(JIS との連携がある英文作成の文献です)

プログラミングは、パソコンを擬人化して語り掛ける文書作成です。新聞・雑誌などは、不特定の大衆向けの情報の編集と出版ですが、実用文書として作成します。論理的な文書構成では、**本体**の作成が最も重要です。**段落**(paragraph)単位の集合で構成します。英文では、段落の物理的な区切りは**改行**位置であって、明快です。一単位の段落は、一単位の論理的な説明に使うことが原則です。複数段落の集合に、章・節・項に順番号を付けた見出しを使うこともします。段落の構成規則には、漢詩の作成で使う「**起承転結**」が多く紹介されています。実用文書では、「転」を省き、「起承結」の三構成が標準です。論理学では、**三段論法**があります。段落単位の文書量は、標準的には、日本語文書では 400 字程度です。英文では文字幅が漢字幅(全角)の約 1/2、単語(ワード)間にスペースを挟みますので、それを含めて1ワードを 5 字で数え、160 ワード、字数にして 800 字程度です。頭の項目は、表題・作者名・日付などです。**目次**(table of contents)、**概要**(あらまし・抄録など):を含めます。概要の文書量は、段落単位とほぼ同じです。簡素な三部構成に圧縮した書式を持たせています。**抄録**(abstract)と似ている**要約** (**ダイジェスト**:digest)は、一単位の読み物として別に編集することがあります。教科書の教材を編集するときは、見本と説明とを兼ねて、独立した文書作品を順不同に集めることがあります。英語や国語では、**読本**(とくほん;リーダー、reader)と呼ぶ教材がそうです。全体を通す論理的な骨格のない、雑学的な文集合です。使い方次第で、一般的な教養を深める話題を提供する材料にはなります。終わりの項目は、索引・付録・書誌事項などです。謝辞、個人的なコメントはここに書きます。書籍のスタイルでは、用語**索引**(index)も、ここにまとめるのが標準ですが、筆者の文書では頭の項目に入れました。

1.4 パソコンを利用する編集

1.4.1 パソコンを介して作文する

「読み・書き・ソロバン」の、大人向け教育も重要です。ソロバンに替えてパソコンが入ると考えることができます。他人に説明する目的で実用文書を作成する作業は、知的な活動です。感情移入をしない作文です。日本語用のワードプロセッサが利用できなかった時代は、原稿用紙に手書きしました。印刷物に作成する作業は、**編集**(edit)と**組版**(植字)を専門とする職業集団に依頼しました。パソコンの利用は、その全体を通した作業をすることになり、**DTP**(desk top publishing:**卓上出版**)の時代なりました。このことは、それまでの編集・組版の職能集団に依頼していた専門知識と技術とを、個人単位で習得する必要が生じました。何を目的として原稿の作成作業をするかの全体を理解した上で、作業を計画します。パソコンを利用するためのプログラミングは、その一種です。全体を総括した表題などを最初に挙げ、それから細かな**部分**の説明に入るように構成します。作業は、部分的な原稿を集め、読み易く、見易くなる順に構成します。これには常識的な定形が有ります。一方、文書の編集手順をまとめたコンピュータプログラミング言語があって、**ML**(markup language)と言います。**編集記述言語**と訳します。インターネットで利用する文書は、**HTML**(Hyper Text Markup Language)形式で作文します。この編集用語に、段落に区切る**タグ**<p>があります。「p」が、paragraph の意味です。文書全体を通信で送受信することが増えてきましたので、Web 版の作成技法も必要な知識です。Web の文書は、段組み表示・フリガナ表示・ページ区切りもしませんので、体裁も別に設計する必要があります。因みに、米語では、**hyper** を接頭辞として使うようになりました。元はギリシャ語です。英語では特大の意味の **super-**が普通です。日本語に訳すときは「**超**(ちょう)」を当てます。超伝導・超音速・超高速鉄道などの用例があります。**極超**を hyper の訳語に使うことがあります。なお、野口悠紀雄の著作(1995)に「超 勉強法」と使われたことから、超の漢字が流行語になりました。因みに、super-の対義語の一つが infra-です。カタカナ語のインフラとして使われていて、基礎・基盤・低位の意味です。赤外線(infrared)、超低音(infra sonic)があります。

1.4.2 字形の有る字と無い字

文字は、眼で見て確認する字形の最小単位です。文字と言うとき、日本語では曖昧な使い方(定義)になっています。英語では、**ラテン文字**(**letter**)、**数字**(**digit**)、**記号**(**symbol**)の区別があり、この全体を **character** とくくります。英語の letter は、**表音文字**です。日本語では**仮名**(かな)が表音文字ですので、英語での説明は Kana letter とします。漢字は意味も持たせますので、Kanji character と言い分けます。機械式の英文タイプライタでは、語並びの間に一文字分の**空白**(スペース)を挟みます。キーボードの下縁にスペースバーがあって、プラテンを一文字分ズラします。活版印刷の組版では、隙間用に活字幅相当の**込め物**を挟みます。行間にも隙間を入れて読み易い印刷物にします。活版印刷の組版は、紙面を活字で埋める作業です。活字並びを揃え、動かないように、隙間すべてを込め物で詰めます。パソコンのモニタは、全画面が発光方式のピクセル液晶画素で埋まっている構造です。地の色が黒になるように工夫して製作されていて、電気信号が加わると明るく光ります。白地の紙に黒で文字を印刷するように表すときは、全画面を明るくしておいて、文字の部分に地の黒が残るような**補色**方式です。字数の少ないテキストファイルのデータ量が少なくても、空白の多いモニタ画面は、空白文字で埋める方式です。パソコンの OS が DOS であって時代、CRT モニタは地の色(灰色)のままにし、文字部分だけが光るようにした表示方式を使いました。文字並びの編集作業は、キーボードからの入力で行います。字形のない制御用のコードがあります。復帰(**CR**: Carriage Return)、改行(**LF**: Line Feed)、シフトキー(**Shift**)、エスケープキー(**Esc**)、タブキー(**Tab**)、バックスペースキー(**BS**)があります。記号用のキーは、機能があっても字形がありません。パソコン用のキーボードは、アスキーコードすべてにキーが割りつけてあります。テレタイプの符号を送信するときに使うキーの集合も引き継いでいます。Ctrl などの追加のキーもあります。

1.4.3 使い方が変った活字

下線(__、**アンダーライン**、**アンダースコア**)は、機械式のタイプライタでは、**重ね打ち**ができました。キーボードに下線用のキーがあっても、単純に横線を引くだけです。そのため、ソフトウエア的な方法で文字並びに下線を引きます。ハイフン「－」は、マイナス記号としての使い方以外には転用できません。そこで、スペースを使う区切りの代わりにも、英字並みの文字として下線を使うようになりました。パソコンの文字入力では、既に文字並びが表示されている文字間に、文字を挿入(insert)し、その文字以降をズラスことができます。これを**挿入モード**と言います。この方式が標準です。バックスペース(back space)キーは、先行している文字を消す機能に代わりました。**上書きモード**(重ね打ち)も選択できます。逆に、後続文字を消して前に詰める機能のキー(delete)があります。ワードプロセッサを使い、また、英語用では逆スラッシュ「＼」の半角文字があります。日本語では半角の「¥」の表示に変りますので、注意が必要です。論理演算子にも特殊な文字種を当てることができるようになりました。

1.5 プログラミング言語の環境

1.5.1 道具・技法・技能

プログラミングは、作文技術である、と捉えます。技術は、実用と向き合い、三つの構成要素「**道具**・**技法**・**技能**」を考えます。道具は、眼に見える形のある物(もの)です。後の二つは、感覚的・経験的にしか理解できない事(こと)です。和語では**ものごと**、漢字では**事物**と当てます。パソコンを使う環境では、「**ハードウエア**(hardware)・**ソフトウエア**(software)・**ユーザインタフェース**(user interface)」に区分します。パソコンを擬人化し、その人に**文**で話しかけ、作業をしてもらう、とする考え方ができました。声で理解してもらう方法は重要な研究課題になりました。。パソコン本体は電気・電子・機械の複合装置です。擬人化した扱いをしても、所詮は道具です。和語には器(**うつわ**:動かない物)と機(**はた**:動く物)の区別があります。どのように動くか、また動かすかの確認の方法が必要です。「始め・止め」の指示も必要です。「何もしない」に当たる命令語(NOP: no operation)があります。犬のしつけでは「待て」があることと似ています。パソコンは、喜怒哀楽のような感情はありません。**五感**(視聴味嗅触)の方は、計測器を介する理解方法があります。そもそも、大型の計算機(メインフレーム)は、軍事用が目的で開発された巨大な数値計算の道具でした。それを**働かせる**(work)、または**機能**(function)させるのは電気信号です。この信号は、眼に見えません。時間的にもその場限りであって、何の実体も残りません。そこで、電気信号を文字並びに代える作文技術が必要になり、逆向きに、文字から電気信号に代える方法も工夫します。その文字並びをファイルに保存しておけば、再現利用ができます。道具としての**テレタイプ**、そして、付属の紙テープ**鑽孔器**は、大量の用紙を使います。この節約には、CRT モニタとファイル装置の利用に代わりました。当初は、文字の表示だけに使いました。電気信号を文字に代えて、オペレータとの対話が工夫されました。文字は図形でもありますが、文字以外の映像を使う**グラフィックス**の利用は、1980 年代からです。プログラミング文書は、コンピュータを**機能**(function)させることが目的ですので、閲覧や保存を考えた実用的な印刷文書に作成することはしません。そのため、かなりの数のプログラミング資産が消滅しています。ただし、COBOLは、英文の**公用文**の書き方に沿う書式を持たせていて、管理用文書として使うことも考えられた言語です。

1.5.2 プログラミングは英語の方言で書く

パソコンの利用を目的したプログラミング言語は、英語を使うことが標準です。文字種は、米国基準の ASCII.コードを使うことが標準です。特殊な文字コードは、データ表記には使います。また、文法的に異なる語彙、また自国語で書く、日本ならば、漢字と仮名とを使う、などの種々の言語違いに対応させることは、**オペレーティングシステム**(OS; operating system)を変えることで行なわせます。漢字は、ラテン文字並びよりも字数を少なくできます。**アイコン**(icon:絵文字)よりも詳しい情報を持たせることができる利点もあります。日本語の環境では、**言文一致**が薦められてきました。公用文では、眼で見て漢字を確かめながら読む**文章口語体**を使います。これは、話し言葉と書き言葉の違いでもあります。ほぼ同じことが、フランス語と英語との違いです。フランスでは、**アカデミーフランセーズ**が古く 1635 に設立され、誰にでも理解可能な言語に純化し、統一することを目的として辞書と文法書の編纂を重要な任務としています。フランス人は、正しいフランス語を使うことにこだわりを持っていて、外来語の利用を抑えます。一方、英語は他言語から名詞の形で取り込む語が多いのが特徴です。これは、書き言葉に向きます。日本でも、新しい用語を使うことに、抵抗を持たない傾向があります。漢字で熟語を造語することは、欧米文化に学ぶときに多く利用されています。マイコン・パソコンの様な、カタカナ用語のスラング化も多くなりました。

1.5.3 インターの付く用語

インタフェース(interface)は、直訳すれば**顔合わせ**です。二つ以上の電気電子装置間をコードやコネクタを使って接続する意味の専門用語であって、漢字では**界面**としていました。接頭辞の inter が付く**インターネット**(internet)は、情報通信の新しい技術用語です。古い英和辞書にはありません。ネットは、電話線などで相互に通信ができる、論理的には網状の構成を言います。**ウエブ**(web)は、蜘蛛の巣を言う名詞です。物理的には網目構造になっていますので、net と同義語で使います。因みに、ウエブは、橋梁工学では斜めの部材を幾何学的に多重の X 形に組み合わせて、桁の腹部を構成するときに言う専門用語です。**ラティス**(lattice)とも言います。それを鋼鈑で置き換えた構造でもウエブと当てる使い方をしています。inter を使う語には、**国際**(インタナショナル:international)があります。また、カタカタ用語ではインターハイ・インターカレッジがあります。高校間・大学間の交流スポーツ会を言う用語です。**学際**(interdisciplinary)の用語もできました。学問の専門間の垣根を外し、自由な研究環境を構成する場面を言います。**情報学**(information science)は、**図書館学**(library science)の一分野であって、文科系に属する専門です。**情報処理**の用語は、今では典型的な学際分野として認められるようになり、文科系・理工系共に交流する分野に進化しました。

1.5.4 文を介して対話する

パソコンを使う環境では、文の編集を、単純な**テキストエディタ**で作業します。テキスト原稿は、ファイルとして保存します。**ファイル**は、元々、事務処理の用語です。欧米企業では、タイピングに高度な技能を持った女性秘書がボスに仕えて文書作成に当たりますので、事務処理の用語を転用している例がかなりあります。CC(Carbon Copy)は一例です。カーボン紙を挟んでタイピングして、手元に残すコピーを作成することです。**電子メール**で利用する項目名になりました。マイクロソフト社は、女性秘書の要望を反映するように、ユーザインタフェースに多くの技術開発をしてきました。以前は、ヒューマンインタフェース(human interface)、マンマシンインタフェース(man machine interface)と言いましたが、年代違いや、性差(ジェンダー:gender)のない user interface に落ち着きました。普通の電卓は、数字・演算子・記号・文字などを表示したキーを押すことで計算を実行させます。キーを押す順序でキーに表示された字形を並べて書くと、電卓が理解できる数値計算の疑似的なプログラム文になります。声に出す使い方はしませんが、ソロバンの**読み上げ算**は、音声による算法指示です。最初に「ご破算で願いましては」と言って、レジスタのクリアを促します。電卓にはクリアキーがあります。続けて、例えば(13+25) の足し算をさせるとき、「1,3,＋,2,5,＝」の順で文字並びをキー入力すると、表示窓に「38」が出ます。窓は、数値の確認用です。この数式文字並びを、声に出して読むとき、日本語ならば、素直に「じゅうさん・足す・にじゅうご・は」と言います。ところが、英語では、13 を thirteen と言いますので、読みが数字の入力順(3 が先、10 が後)と逆です。式を文字並びで表すとき、命令形を使う文形では、動詞を先行させて(Add 13 and 25)とするか、平叙文では、仮の主語を最初に立てて、(C is equal to 13 plus 25)のように書きます。電卓の使い方は、欧米人の言語習慣である SVO 順とは違いますので、大問題になるのです。算法については、**ポーランド方式**・**逆ポーランド方式**の用語が、また、**左辺値**・**右辺値**・**代入文**の用語も必要になります。

1.5.5 システムの考え方が必要になった

システム(system)は、単・複の区別がある普通名詞です。集合名詞の性質もあります。用語としての説明は、「**複数**の要素の集合体で**一つ**の機能を持つ**もの**」とあります。コンピュータは複雑な装置です。多くの部品を総合して構成し、それを使うための**人**も専門違いで多くの人が参加します。そのため、**システム**に組み上げ、関連を持たせながら、一つの目的を達成するように制御します。その最高位置にある管理者を、systems engineer と言います。複数形の systems であることに注意が必要です。日本語の用語では**システムエンジニア**の誤訳が使われています。単数形の system engineer は、或る特定した機械装置の作業に当たる技術者の意義で使います。システムは、機械装置だけでなく、人の集合でも言います。日本語では、組合・組織・系などがそうです。パソコンでは、Operating System を使います。これは一つのプログラムですが、複数のデバイスを効率的に利用することが目的です。複数のシステムを合わせて、より大きなシステムに構成するときに、複数形の systems と使います。日本語の環境では、名詞の単複の使い分けに神経質ではないこともあって、説明不足・誤用・誤解が多く見られます。プログラミングの場面では、専門ごとの要望を考えて、ユーザ・専門家・プログラマなど、人の集合でシステムを構成するのが合理的です。企業がパソコンシステムを構築する場合には、業務に特化した仕様を必要とします。企業秘密があるときは、信頼関係のあるソフト会社に依頼するか、自社開発のチームを編成します。このシステムを利用するには、新入社員などを主な対象とした教育過程が必要です。学校教育では、毎年、多人数の新人が入学してきますので、教授法が研究されます。しかし、成熟した企業では、新人の入社数が学校に比べれば低い率になりますので、新人教育をシステム化して組み込むことが重視されないことがあります。新人は、先輩を見ながら、見様見真似で、悪くすると、偏った経験を積みます。いじめの意識は無いのですが、結果的にいじめになることも起こります。このことは、企業の IT(information technology)の実力が、国際的に見て低下を招く原因になり、社会問題化してきました。

1.5.6 プログラミングもしなくなった

我々の普通の生活環境では、パソコンを使う知識が必要にはなってきました。しかし、専門化が進み過ぎて、一般の人は、作文に当たる、プログラミングまではしなくなりました。何かの情報を得たいとき、**インターネット**で検索すれば、先人が努力して作成した、資料が見つかる時代になりました。外見(そとみ)には、科学技術の進歩ですが、裏からみれば、閉鎖的な環境が増え、対面で対話に参加することで得られる IT 技能(information technology)の伝承が途切れる事態が進行しています。全くの初心者がプログラミングを覚えるには、BASIC などの簡易な言語で、対面会話を交える**実習**の過程が必要です。技術開発は、視線で言えば未来を向いていて、これから実行させたい手順を書く作業です。正しいプログラム文を書くことを覚えるため、プログラミングを勉強しなければならないのです。パソコンで文書を扱うとなると、パソコン本体についても常識的な知識を持っていることが必要です。この接点が**レジスタ**と**ファイル**と言う概念です。この解説は第二分冊にまとめます。

1.6 プログラミングの説明に使う用語

1.6.1 書式と文とで構成する

プログラミング文書本体に使う用語ではなく、説明書・入門書・マニュアルで使う英単語は、別に理解するか、憶えておく必要があります。英語が話されている環境で使う日常用語は、説明が抜けます。厄介な語は、頭字語(acronym)です。元の綴りが分かっていることが前提の語です。例えば、PC は、personal computerのことですが、建設関係では prestressed concrete のことです。日本語ではさらにカタカナ語の短縮化が使われます。マイコン・パソコンなどがそうです。因みに、語尾が「…コン」で終わる語は、広辞苑の逆引き辞書で引くと、漢字の場合も含めると、約 300 語もあります。

実用文書の中身は、**通知文**(おしらせ)または**依頼文**(おねがい)です。役所からくる文書には文語調、命令調の文もありましたが、最近は優しい文体になりました。文を並べる約束が**書式**(format)です。プログラミング文書では、通知文は、**非実行文**、依頼文は**実行文**と言い分けます。依頼する内容を、プログラム名の表題として使います。**英語風**で作文しますので、丁寧に言うときは、「English-like-structured language でプログラミングをする」と言います。ハイフンで繋いだ 3 語は、**複合形容詞**です。プログラミング文書では、文単位を statement または expression と言い換えます。プログラミング文書はパソコン側に理解してもらうことが目的です。書式と文の書き方には注意しますが、デザインなどに工夫をかける**様式**(style)は重要としません。

プログラミングでは、パソコン側で理解する語が決めてあって、**予約語**(reserved word)、または**キーワード**(keyword)と言います。この区別は、言語の説明書によって定義と解釈が異なります。COBOL は、予約語が 500 もあるとされ、逆に FORTAN は予約語を使わない言語である、とされています。予約語は文法的に主に動詞と名詞です。関数名または処理名は名詞です。実行文は、目的語を従えた他動詞の命令文の集合で表しますが、**道草**を食う処理も入ります。**繰り返し**文がそうです。関数やサブルーチンの計算も道草処理です。

一単位の実行形式のプログラムの作成は、実践的には、複数のテキストファイルに分けて作文します。個別のファイルの書式は、見出し(ファイル名の方)を付け、段落を考えた文集合に構成します。実行時に、最初に制御が入る場所を明示的に表すファイル名は **MAIN** です。他のファイルは、処理単位名か関数名を付けた副プログラムです。普通の文書では、文並びを順に読み、後戻りも読み飛ばしもしません。プログラム文では、この構造を**順接**(sequence)と言います。加えて、**分岐**(section)と**反復**(repletion)の実行があることが特殊です。これは、別の場所の文に制御を移します。単純な方法は、文に見出し(**ラベル**)を付けておいて、**GOTO 文,JUMP 文**でその場所に制御を移します。これが、実行手順を混乱させますので、この形式の文を使わないようにします。分岐と反復の部分をブロック化しておいて、処理が見掛け上、順に流れるようにします。これを**構造化プログラミング**(structured programming)と言います。対義語が、俗称ですが、**スパゲッティプログラム**です。この考え方に沿わせるため、**宣言文**を先に書き、**定義文**は後に書き、中身の実体を決めます。宣言に使う用語は、プログラミング言語では予約語です。代表的にはデータ型名があります。整数型・実数型・文字型などの区別を言う用語がそうです。**コメント文**は非実行文として任意の箇所に挿入できます。読み易さを助ける様式にも使います。ただし、コンパイラでは何がしかの処理をしますので、書き方の文法があります。

宣言(declaration)と**定義**(definition)との対は、あらゆる場面で表れます。目次・索引・カタログは、宣言の性格があって、実体がどこにあるかの情報を知らせます。プログラミング文書では、宣言に使う予約語の説明は、マニュアルの方に含めます。データの型名が代表的な宣言名です。FORTRAN では、英字の I～N で始まる名前は整数型の変数であると約束してあるのが、暗黙の宣言です。宣言の段階では具体的な実体をまだ指定していませんので、プログラマ側は具体的な変数名や配列名を付けて、実体を宣言します。宣言と定義との説明には、例えば、日本語で「タマは猫である」と言う表し方を挙げます。「猫とはこう言う動物だ」という了解があります。これが暗黙の宣言の利用です。「三毛猫」は「猫」から**派生**した宣言と言います。論理学では猫の宣言の**外延**と言い、プログラミングの用語では**継承**(inheritance)という難しい用語も使われています。例文は、猫の宣言を踏まえ、或る特定の猫に「タマ」とか「ミー」の名前を当てるのが定義です。宣言と定義とで作成されるデータ集合単位は、学校で使う**クラス**(class)の用語を当てることもします。**もの**(物:オブジェクト)と解釈すると良いでしょう。この**もの**の複製を**インスタンス**(instance)と呼びます。英語で、for instance(一例を挙げれば)で使われる用語です。つまり、宣言を設計図とすると、定義で複数の名前違いの実体を作成します。プログラマは、同名の猫が居ないように命名を工夫します。別の場所で、同じ名前で、別の実体を区別することもあります。これは混乱を招きますので、これを避ける考え方が**スコープ**(適用範囲:scope)です。原義は、狭い範囲を見る、標的や狙いのことです。動物で言えば縄張り範囲、眼でみて観察できる視界、のように使います。猫を飼っている家の名前などで修飾する方法を提案している言語もあります。

1.6.2 パソコン側内部で使う語

パソコンを擬人化して考えるき、パソコン側にも**母語**があるとします。元は電気信号の**命令コード**(**オペレーションコード**)です。CPU の製作会社別で違いがある方言です。複数のバイト並びで機械語を構成します。人にとっては全くの暗号ですので、意味が分かる英数字並びのニーモニック(記憶記号)を人の方で決めます。普通の電卓は、キーを押せば、内部的に命令コードが送信される方式のユーザインタフェースです。キーの表面に文字・数字・記号が書いてありますので、それを並べて書けば、最も単純なプログラミング文書になります。ユーザインタフェースは、種々の工夫がされてきました。普通の電卓のキーの数は 30 程度、関数電卓でも 50 程度です。予約語の品詞は、基本的には名詞と動詞です。C/C++言語では**トークン**(token)と言い換え、区切り符号も含めます。FORTRAN では、予約語を使わない言語であるとの説明もありますので、誤解が生じています。正確には、代数式を書きますので、「命令文を使わない」と言う意味です。COBOL は飛び抜けて予約語が多く、500 程度もあります。C 言語の予約語は 32 語、C++では約 40 語が追加されています。データベースでは、検索に使う大量の予約語を**シソーラス**(thesaurus)に編集しておきます。ロジェ(1779-1869)が命名した類語辞典です。ユーザがキーワードを決めるときの参考にします。**上位語**(UT; upper term)、**下位語**(LT; lower term)、**関連語**(RT; relational term)の区分を集めた編集です。

1.6.3 キーワードの意味

プログラミング言語で言うキーワードは、プログラム名、コマンド名、関数名、処理名、型名です。或る長さの文字集合の**単語**です。**コマンド**の場合は、引き数を必要とする**命令文**の形で使います。MS-DOS のコマンドは、便利な使い方ができますので、Windows の環境でも利用できます。関数名は、整数型・実数型のような**型名**で分類します。キーワードは、数も種類も多いので、別のマニュアルにまとめられていることが多く、何が有るかが直ぐには分かりません。FORTRAN の組み込み関数名はキーワードの方です。システムが使うキーワードは、ライブラリにまとめてあって、そこに実体のコードがあります。中身がどうなっているかも分かりません。C/C++言語では、ライブラリのファイル名を**インクルードファイル**名としてユーザプログラム文の頭の部分に追加の宣言を必要とします。プログラミング言語は、感情を表す形容詞や副詞を使いません。大小・多少・長短などの程度や比較を表す演算子用の語はあります。五感(視聴臭味触)は、計測方法があって、デジタル化ができます。トークンは品詞と同義です、文字並びは、英字(letter)・数字(digit)・記号(symbol)に分類し、その総合は、**character set** です。また、**letter se**t は、**フォント**(font)の用語の方を使い、デザイン的な要素を持ちます。

1.6.4 ユーザ側がパソコン側に知らせる語

擬人化したパソコンは、我々と音声で交流するのではなく、ビット、さらに 8 ビットの並びのバイトを母語としていると考えます。プログラミング本文では、英字・数字・記号は、英文用キーボードにある半角文字、それも、先頭ビットが 0 である7ビットのコード(ACII コード)しか使いません。8 ビット長さの文字は、拡張文字コード扱いです。日本語用キーボードは、全角文字(2 バイト文字コード)の使い方ができる欲張った仕様ですが、使い方に混乱も起こります。プログラミング本文に使うと、エラーになります。人の方で使う英語の文字種は、機械式または電気機械式電信機のタイプライタに在る文字で制限を受けました。6 本の針金の組み合わせで、活字を付けたバーを動かして、タイピングをするメカニズムでした。ビット並びの 6 単位を拡張して 8 単位にした符号がバイトです。バイトの使い方は、3 通りあります。第一は数値、第二は半角文字コードです。数値と数字とはビット並びが別定義です。第三が、プロセッサに指示する**命令コード**(オペレーションコード)です。普通の電卓では、加減乗除のキー(＋－×÷)を押すと命令コードがレジスタに送信されます。パソコンの場合には、他動詞の機能を持つ、複数のバイト並びの語句です。その集合で、或るまとまった処理に組み立て、処理名(キーワード)をユーザが命名してパソコン側に知らせます。変数・関数・処理などに付ける固有名を、総称で**識別子**と言います。原則として、英字で始める英数字並びです。記号は使いません。予約語とキーワードも使うことを避けます。

1.6.5 英語流に単語を並べる

英語は、単語の並べ方で意味が決まります。典型的な語順は、主語・動詞・目的語の順であって、これをSVOの順と呼びます。日本語の場合には、名詞に助詞の「てにをは」を付ければ、ある程度順序が自由ですし、動詞も活用しますが、動詞が文末にくるSOVの順が標準です。コンピュータ言語の構文には、コンピュータに対する英語の命令文の形を取った、VO、または VOO の形が用いられます。典型的な用法がコマンド文です。文頭の単語が名詞であっても、動詞的な意義を持ちます。例えば、整数変数を宣言するとき、「int i;」のような表現を使います。日本語でこの意義を説明するとき、語順を変え、助詞を補って、「i を整数とする」のように読み換えて理解します。漢文(中国語)の語順も英語と同じSOVですので、日本人が漢文を理解するときには「レ点、返り点、番号」を付けて日本語の語順に直して理解しています。

1.6.6 定義文・宣言文は命令文の形式で書く

宣言文の例として「タマは猫です」を挙げました。「タマ」は、ユーザが命名した語(定義)、「猫」はシステムが宣言した語です。プログラミングでは**変数**であって、型名は文字型です。これは日本語の構文です。英語風の表現は、日本語の習慣である助詞がありませんので、語の並び順で定義文を構成します。「cat tama」のように逆順です。つまり SVO 形式の「tama is cat」の文の意義ではありません。宣言を表すときの予約語は、C 言語では char, double, float, int, long など、C++ではこれに bool が追加されています。宣言をするとき、修飾に使う予約語が幾つかあります。C 言語は auto, static, extern、C++では private, public などがあります。実行文は、**他動詞**の命令形を使い、**直接目的語**(オブジェクト)を従えます。これを argument と言い、日本語の用語は**変数**または**引き数**です。**間接目的語**は、日本語の助詞では、「…に」を後置詞に持つ語です。英語では、前置詞を使わない構文は、間接目的語・直接目的語の順です。日本語でも「…に、…を」の順が普通ですが、逆順に「…を…に」と言うこともできます。自動詞は、パソコン側が勝手に動く意義を持ちます。したがって、自動詞の go, move, run を使う場面では説明が必要になります。例えば、前置詞を添える goto, moveto の語を定義しています。run は、他動詞 execute であると解説しているコンピュータの用語辞書があります。**宣言文**(declaration)・**注釈文**(comment)は、**非実行文**です。非実行文でも、パソコン側では何らかの判断と処理とが必要です。代入文の形にするとき、命令形の動詞 Let を文頭に使います。なお、日本語では、自動詞を受身形で使う場面があります。非常に丁寧な敬語表現の場合と、迷惑を表す場合とがあります。相手の動作が、こちらで制御できない場面で使います。

1.6.7 代数関数のプログラミング

代数式、例えば「A=B+C」の表現は、数学的にはイコール記号の左右が等しいことを意味しますので、「B+C=A」と書いても間違いではありません。しかし、コンピュータ言語では全く意義が異なります。前者の表現はSOVの語順「A is equal to B plus C」を記号化したものですので、式そのものも、そのように読みます。日本語でならば、「B 足す C は A になる」が自然な言葉遣いであって、動詞が最後にきます。コンピュータの処理手順は、日本語の語順で実行されます。この式の実行は、(B+C)を先に計算しておいて結果を A に代入します。コンピュータのコンパイラは、英語流の語順をコンピュータ処理順に直す翻訳を内部的に行なっています。そのため、イコール記号を使う表現を**代入文**(assignment statement)と言い、(B+C)の部分を**式**(expression)と区別します。両方繋がったものを**式文**(assignment expression)と言い換えることがあります。

メインフレームは、数値計算に利用することが当初の目的でした。それには、普通の代数関数(sin, cos, log, …)を、高速で精度良く計算するプログラムが必要です。1950 年頃は、機械語でプログラミングをすることが主要な研究課題になっていました。一通りの研究が終わって、FORTRAN では組み込み関数の扱いになりました。関数電卓でも利用できるようになって、それまで、理工系の図書としてベストセラーであった代数関数の数表出版が全滅してしまいました。代数関数の数値計算の**アルゴリズム**の解説は、教育科目として重要であったのですが、今では眼にすることが少なくなりました。これは、基礎知識を習得する場面に、大きな穴が空きます。外見(そとみ)には高度な技術の進歩であっても、その裏に生じている深刻な問題です。

科学技術関係の書籍では、数式を使います。活版印刷が主流であった頃、編集・組版に特殊な技能が必要でした。数式は、分数表記のように複数行に渡ることもあり、特種な記号、例えば積分記号なども使いますので、プログラミングでは画像(オブジェクト)扱いをします。その編集言語の一つが TeX です。数式を元にして数値計算をするには、プログラミング言語で書き替えをしなければ実用的な計算ができません。この方法を研究する学問分野は応用数学であって、**数値計算法**として、大学では工学部が主な研究拠点です。

コンパイラ型で編集するプログラミング言語は、テキストファイルの**ソースコード**で、論理的に独立な実行単位にコンパイルします。作業性を考えて、複数のファイルに分けることもします。このとき、別のファイルの関数や変数を参照するための約束が必要です。文章化の書式について言うと、英文は文単位の区切りはピリオド(.)ですが、C/C++プログラム文単位の区切りはコロン(;)を使います。コンパイラは、プログラム本文の一つ以上の空白を論理的には一つ扱いとし、タブも改行コードも一つの空白扱いです。したがって、プログラム文書単位は、文の集合であって、ファイルの始めから終わりまでが一続きです。しかし、見易さ(体裁)を考えて、文の切れ目(;)、論理的な文字の切れ目(トークン単位)で物理的な改行することは自由です。標準的な体裁は、一行の文字数が半角で約 80 字以内、適度にタブを使った字下げ(インデント)をして、論理的な文単位を分かり易く表します。この体裁は、コンパイラ付属のテキストエディタが、半ば自動的に調整してくれます。切れ目があっては困る長い文字データを扱う場合には、物理的な改行で切れ目が入らないように、論理的に継続することを示す行末の記号(¥、ASCII コードの字形は逆スラッシュ)が決められています。

1.6.8 プログラミング言語開発の概略史

年代で言うと、戦中から 1950 年頃まで、数値計算を主目的とする大型コンピュータ(**メインフレーム**)を制御する**オペレータ**は、紙テープ装置付きのテレタイプを使って、データの入出力に当たりました。メインフレームは、巨大な施設でした。機械のような外観を持っていました。人の方で決めた入出力コードを**機械語**(マシン語)と呼びました。人が読むとなると、全くの暗号並びです。そこで、この意味が分かり、**記憶**用単語(**ニーモニック**: mnemonic)を考え、これで命令語を書けば、翻訳して機械語を生成できる方法が工夫されました。これが**アセンブリ言語**(assembly language です。**FORTRAN** は 1954 年に発表され、**高水準言語**の始まりです。マシン語・アセンブリ言語は**低水準言語**と差別する言い方ができました。**COBOL** は、1959 年の発表です。プログラミング教育に利用することを考えた **BASIC** は、FORTRAN を読み易い文書スタイルにした言語であって、1960 年代半ば頃に発表されました。**UNIX** は、マシン語によって 1969 年に開発された**オペレーティングシステム** (OS)です。マシン語と相性の良い **C 言語**が 1972 年に発表されたことを受けて、1973 に C 言語で書き替えられ、その後の各種の OS やプログラミング言語の開発に応用されています。**C++**は 1983 年の発表です。C 言語の改良版の顔を持ちます。ここまでの年代は、32 ビットまたは 36 ビットの **CPU**(Central Processing Unit)で機能するメインフレームの時代でした。8 ビットの**マイクロプロセッサ**を採用した**マイコン**(my computer ではありません)NEC-PC8001 は、1979 に発売され、これがパソコン(パーソナルコンピュータ)時代の始まりになりました。

メインフレームを主力製品としていた IBM 社がパソコンの開発に参入したのは 1981 年からです。1984 年に発表された IBM-PC/AT は、16 ビットパソコンの開発モデルとなり、多くの企業がパソコン開発に参加しました。とりわけ、NEC-PC9800 シリーズの 16 ビットのパソコンは、1982～2004 間に 1800 万台強も生産され、多くの**アプリケーション**(application:コンピュータプログラムと同義)の開発に貢献しました。しかし、16 ビットのパソコンの利用が大衆化するにつれ、メインフレームと比べると、機能的な不十分さが問題にされるようになってきました。32 ビットのプロセッサを搭載した実用的なパソコンの発表は、**DOS** の環境とは全く違う OS:**Windows 3.1**、に 1993 年に移行したことが始まりです。多くのアプリケーションは改版を余儀なくされたのですが、2000 年頃には、16 ビットパソコンの利用時代に有った消化不良的な課題がほぼ解消されるようになりました。ソフトウエア的な改変である OS の変更と、ハードウエア的な改変である、CPU(Central Processing Unit)のレジスタ(または **MPU**: Micro Processing Unit)のビットレジスタ数が 16 から 32 に変更されたこととは、アプリケーションの開発環境を大きく変えました。2010 年頃から、64 ビットのパソコン時代に入りました。

プログラミング(作業)は、一単位の実行形式のプログラムの作成です。幾つもの別**プログラム**が作業過程で使われます。(1)**テキストエディタ**を使ってテキスト形式の**ソースコード**を編集する;このソースコードを、(2)**コンパイラ**(compiler)を使ってコンピュータが理解できるマシン語に翻訳して**オブジェクトファイル**に作製する;(3)**リンカー**(linker)を使って幾つかのオブジェクトファイルを論理的に繋いで実行形式プログラムに組み上げます。この作業全体は、使用するコンピュータシステム(**オブジェクト・コンピュータ**)と関わります。これが(4)**OS**(処理系)です。作成したプログラムコードが設計通りに機能することを確認するのに**デバッガー**(debugger) (5)を使います。この作業全体を管理するツールが 1993 年頃、ベンダー(vendor:販売会社)から**統合開発環境**(**IDE**: Integrated Development Environment)として発売されるようになりました。例えば、Borland 社の C++Builder5(**2000**)が代表的な IDE の一つです。ユーザがプログラミングを覚えることの大半は、プログラミング言語のベンダーが提供する IDE の使い方に費やされています。

コンピュータは、何の目的にも利用できる汎用性を謳うこともありますが、COBOL や FORTRAN のように、専門別に固有のプログラミング言語を開発する方向に進んでいます。Windows の OS が標準なりました。このモニタ画面は、ボタンなど部品要素の図柄、アイコンなどの絵文字を表示して、それにカーソルを合わせてクリックすることで、疑似的に装置を操作する方式が使われるようになりました。この画像(グラフィックス)を、代名詞的にオブジェクトと総称し、オブジェクトを利用するアプリケーションの作成を、**オブジェクト指向プログラミング**(object-oriented programming)と言うようになりました。プログラミング言語の COBOL は、「Common Business Oriented Language」の頭字語から命名された名前です。ここに既に、oriented が使われていました。パソコンを使うプログラミングは、「何をしたいのか？」の目的意識を、はっきりと始めに持つことが重要である、の認識が育ってきました。それも、具体的であればあるほど、明確になります。OS が Windows 系の **GUI**(Graphical User Interface)の方式を採用するようになって、グラフィックス指向(利用の意味)のプログラミングが多くなりました。

1.7 説明不足の用語

1.7.1 英語の環境を大きく受けている

日本の和語になかった外来語は、古くは中国語、明治以降は英語を始めとした欧米語を言い、名詞として取り込まれます。この名詞を別の品詞で使う場面が起きます。「**スル名詞**」は「…する」と動詞に使い、「**ナニ名詞**」は形容詞または副詞に使います。例えば、「…優雅な(形容詞)」、「…優雅に(副詞)」です。これは、日本人の発想ではなく、外国人が日本語を勉強するときに使うようになった用語です。語幹の部分は、原語でも動詞・形容詞の意義を持つ語であるのが基本です。英語の辞書では品詞分類がありますが、漢和辞典には品詞分類がありませんので、使い方に揺れが起こります。近年では、電子レンジを使う場面で、「チンする」の動詞なども見られます。ただし、俗な言い方とされています。

コンピュータの技術は、主にアメリカ(米国)主導で開発・研究がされてきましたので、日本で発売されている参考書は、英語版を元にしているのが殆どです。そのため、日本語には無い用語を使う場面があり、また、同じ意味と思われている語の概念が、異なることがあります。とりわけ、英語環境に居る人には常識的な日常用語の説明が抜けます。また、社会環境の違い、文化の背景違い、宗教がらみの考え方の違いもあります。技術用語には、特殊な語も多く、また日常言語とは異なった意義を持つ使い方もありますので、日本語では**業界用語**と区別することをしています。英語の環境でも専門用語辞書(dictionary)があり、また説明書(glossary)があります。パソコン用語の幾つかを、この節で説明します。

1.7.2 言葉遣いと言葉使い

言葉遣いは、相手に対して、声で伝える言い方のことです。乱暴な言葉遣い、丁寧な言葉遣い、のような区別があり、感情移入があります。一方、言葉使いは、事務的な場面での文字並びの約束の意義があります。声に出すよりも、文書での使い方が主になります。書式とは、適切な用語を選び、語順の決め方の約束です。**推敲**は、漢字の推を選ぶか、敲を選ぶかの判断を決めるときの熟語です。プログラミングは、擬人化したパソコンに文書で話しかける、英語風の言語を使う事務的な作文です。基本的な語彙数は 100 程度です。プログラマは、データ(物)や処理(事)に名前を付け、文の集合に構成します。最初に、「この名前を使うよ」の**宣言文**を書き、後の行に実体を説明する**定義文**を書きます。データ宣言は、プログラミング言語側では何種類かの**型**が決められていて、例えば整数型(INT)、文字型(CHA)などです。一方、最小単位の基本的な処理名があります。四則演算がそうです。記号として(+ − ＊ /)がありますが、明示的な宣言文は書きません。機械語では、文字で(ADD, SUB, MULT, DIV)などが予約語として定義されています。プログラマは、関数とデータとで構成する処理単位に名前を付けます。これも、宣言文を先に書き、実処理の構文(定義)を後にまとめる書式が標準になってきました。処理名は、プログラミングでは文単位の先頭に置く命令形で使い、目的語にデータ名が続きます。処理の定義名には、代表的には関数名があります。処理が複雑であるときは、副プログラム(subroutine)として、CALL の動詞を頭に追加することもします。しかし、宣言や定義であることを明示的に書かない書式が多く使われます。

1.7.3 プログラム

英語が元である**プログラム**は、日本語化して、**手順**を言う用語として普通に使うようになりました。これに当たる日本語には、**式次第**(しきしだい)があります。入学式、表彰式などの進め方などを書き出すときに使います。プログラムの綴りは、program(アメリカ英語：米語)です。イギリス英語は(programme)と書きます。カタカナ語になったプログラムは、テレビの番組、音楽会、学校行事での運動会など、予定を書きだす文書に広く使われるようになりました。一方、**プログラミング言語**(programming language： mm と綴ります)は、コンピュータを擬人扱いして、その人に理解してもらい、作業をしてもらうときに話し掛けるときの、英語風の**人工言語**です。幾つもの種類があります。これを**機械語**と言います。実体は、電気信号の組み合わせを人の方で決めた符号です。コンピュータ側では、装置内でこれを**母語**として使って情報通信をしているとする考え方を生みました。

我々が日常使っている言葉は日本語です。英語、フランス語など、「…語」と呼ぶ方を、人工言語の対義語で**自然言語**と言います。言語の種類を細かく分けたいところですが、**方言**(dialect)違いも含めると、実際に幾つ有るかは分かりません。コンピュータプログラミングは、人工言語を使って、**実用文書**を作成する作業です。何を目的にして作文するかで、多様になります。何にでも利用できる言語を**汎用言語**と言います。代表的には **C 言語**です。しかし、或る特定の利用目的を持たせた言語が多く開発されるようになりました。数値計算を目的とした FORTRAN、事務処理を目的とした COBOL が、その走りです。初心者向けのプログラミング教育用言語が BASIC です。これらのプログラミング言語を利用して、さらに開発対象を絞り込んだ実行形式のプログラムを、**アプリケーションプログラム・ユーティリティー**などと言い分けます。

1.7.4 名詞の単複区別と冠詞の使い分け

パソコンのプログラミングは、一種の英作文です。英語のネイティブスピーカーならば常識的な言葉遣いであっても、日本人は誤解して書くことが少なくありません。名詞の使い方を日本人が理解できる考え方は、少し苦しい説明ですが、**俳句**の作文と似ているところがあります。俳句は、短い語並びですが、意味の中に**季語**を含ませる規則があります。季語は、予約語の性質を持ちます。**歳時記**は、季語の宣言を集めた辞書です。文芸であっても、辞書のような実用書があることが特別です。日本語の辞書は、国語辞典の表題が付きます。中身は、漢字用語と和語用語との対応であることが多く、言わば、言い換えが集めてあります。例えば、**離島**は、漢語では島を人が離れるの意義です。和語では、送り仮名を付けて離れ島と使うのが紛れないのですが、離島と使うことが普通になりました。もう一つ、**落馬**があって、人が馬から落ちると説明があります。中国語の語順ならば、馬が崖から落ちる意味になるからです。英語の辞書は、「この語の意味はこうだよ」の宣言を集めてあります。動詞・名詞などの品詞区別をすることは、定義です。専門用語は、英語でもコンユータ用語の辞書がありますが、普通の意味とは違った使い方をすることの説明を集めてあります。日本語辞書は対訳で済ます編集が多いので、意味までは分からないことが起こります。英語では、名詞が文中に使われるとき、単複の区別、冠詞の有る無しで意味が変り、意味の使い分けをする規則があります。日本語を使う環境には無い習慣ですので、英語の学習では多くの誤解・誤用が起こります。下の表の dog は、名詞です。5種類の文で使い分けを例示してあります。文例⑤は、日本人には想像もできない意味になる言い方です。プログラミング文書では、名詞に冠詞を使いません。複数は、配列で表します。この約束は、日本人には助かります。

番号	文　例	意　味	解　説
①	I like a dog	名前は言わないが、或る犬が好きだ	猫など、他に動物がいても、犬は一匹です
②	I like the dog	あの例の犬が好きだ	誰かが飼っている既知の一匹の犬です
③	I like the dogs	あの例の犬たちが好きだ	二匹以上飼っている場面が想像できます
④	I like dogs	犬が好きだ	一般論です。猫も好きかもしれません。
⑤	I like dog	犬の肉が好きだ	冠詞の無い単数形は不加算名詞扱いです

「参考文献:マーク・ピーターセン、日本人が誤解する英語、光文社文庫、2010 他」

1.7.5 be 動詞

英語の be 動詞は、主語の人称・時制・単複などの区別でスペル違いの語(is, am, are, was, were,…)の活用形を使い分ける**自動詞**です。**存在動詞**とも言います。**他動詞**を**受身形**に使うときには**助動詞**とみなされます。ドイツ語では sein、フランス語では e^tre です。語学として勉強するとき、真っ先に活用形を覚える必要があります。日本語では、「…です、…ます、…である」に当たる語です。状態を表す宣言と定義に使います。存在状態を表したい場面での使い方に違いがあります。人や動物は「**居**る・居ない、物は「**在**る・**有**る・**無**い」の漢字で区別します。be 動詞は、原形で使うことはありません。シェークスピアのハムレットで「to be or not to be …」は、特例として紹介されています。翻訳が難しい語句としても有名です。ビートルズの歌曲「Let It Be」は「あるがまま」と訳されています。定義や宣言を丁寧に説明したい場面では、**使役動詞**の let, make, を添えます。be 動詞は、常識的な語であることから、プログラミング言語で使うことはありません。これに代わる語が、int, float, など、型の宣言をする**予約語**(reserved word)です。ただし、**キーワード**(keyword)の用語との使い分けは、プログラミング言語ごとに違い、曖昧です。be 動詞は、C 言語に使う例があります。文字種の違いを調べる関数に、疑問文形式の、「isalpha, isdigit, …」があって、include 文の<ctype.h>で宣言をすれば使うことができます。

1.7.6 オブジェクト

英単語の object(オブジェクト)は、日本語では、**ものごと**(**物と事**)の二つを当てます。物は、形、つまり、実体があります。物には人も含め、**者**と使います。法律用語は**有体物**です。電力・電気信号・情報などは、**無体物**とします。事は、歩く事のような**動名詞**を指します。英語では、現在分詞形(…ing)と不定形(to…)があります。自然現象なども言い、**事象**の用語があります。動作や情感なども言います。object は、物と事との区別をしない語であるため、日本語に訳すときに混乱を起こします。例えば、program と言うときは、文書化された物扱いをし、ソフトウエアの扱いをし、programming は、事の方を言い、人が作業する意義のユーザインタフェースの用語と関係します。object は、英文法の用語では**他動詞**の**目的語**を言います。名詞・代名詞のすべてを目的語として使うことができますので、**対象物**の言い方もしています。object の同義語に thing があります。こちらは、漠然とした意味を持たせた日常語の使い方をします。object の場合は、いま、具体的に話題に取り上げているものごとを指す意義があります。コンピュータ言語では、モニタ上に表示する図形要素(直線・矩形・円など)と、アイコンなどの映像、そして図形としての文字も含める用語です。

1.7.7 複合形容詞

ここで、二つの語句を説明例として挙げます。「object-oriented graphics と object graphics」。前者の、ハイフンで繋いで一語にして使う object-oriented が、**複合形容詞**(compound adjective)です。orient を動詞の過去分詞形を使っていますが、受け身または過去の意味ではなく、他動詞としての、使うまたは利用する、の意味です。「モニタ上に幾つものグラフィックス(画像)を使って描く**事**で得られた作品(物)だよ」の意味です。英語の語順では、動詞-目的語が常識ですが、目的語-動詞の順になっていますので、ハイフンで繋いで、「一語だよ」と表しています。動詞の orient に**指向**の訳語を当てたため、訳の分からない用語になっています。一方、後者の用語は、orient がありません。幾つもの画像がある中で「いま話題として取り上げている**物**(画像)はこれだよ」の文脈の中で使います。

物扱いのできる画像データは、ファイルに保存し、再描画ができます。個別に円の位置や大きさなどを変える使い方ができますので、CAD (computer-aided design)に応用されています。ここでも、ハイフンで繋いだ computer-aided も複合形容詞です。コンピュータを使う、と言う意味です。

もう一つの用例を挙げます。何を目的にしてプログラムを作成するか、その「**何を**」に「図形要素を意味して object を当てる、その用語が「object-oriented programming;です。Windows の OS の環境でのプログラミングの紹介をしていた初期のころ、例題に、電卓の設計が使われました。現在では、アクセサリツールのプログラム群のなかに含まれているユーティリティプログラムです。

1.7.8 業務用マニュアルなど

義務教育の段階を始めとして、大学などの高等教育で利用する教科書やテキスト類は、原則として使い捨ての利用で編集されています。参考にする補助教材、例えば、日本地図・世界地図などは、便利であるとして、残しておくこともしています。国語、英語などの辞書類は、多くの種類が出版されていますので、個人的に選択して購入する参考書です。新聞・雑誌類は、読み捨てされます。興味のある記事は、その部分だけを切り取って自前の資料集を作り、残りは捨てます。全体は、図書館などで一時的に保存されることもありますが、普通は年度が替われば廃棄されてしまいます。全体保存は、発行元が本来は責任を持つべきです。国として、国立国会図書館や公文書館(アーカイブ)は、私的な文書も含めて、その保存業務の重要性と機能とが認識されるようになってきました。

大型書店では多くの専門書が扱われています。これらの実用書には二種類あります。一つは「資料集、百科事典類、ハンドブック」などです。電子化辞書や、インターネットでの検索が便利になって、見掛けなくなりましたが、個性的な項目では、書物形式は独特の価値があります。もう一つが専門単位での入門書です。パソコン関連では、「初めての…、分かり易い…、易しい…、入門…」のような接頭辞を付けた書物が並びます。多くの場合、著者が個人的に経験して覚えた経過をまとめた内容であることに注意が必要です。本人が常識と考えていることの説明が抜けるからです。パソコン実物を前にして、できれば、一対一の対話で手ほどきを受ける環境にないと、全く難解な書物になっているのです。

個人、また、或る専門家の集団では、自前の資料集を作ることもします。郵便切手を集めるなどの趣味的な収集ではなく、利用頻度の高い資料です。一ページ程度のコピーの形で手許に置きます。代表的には、企業では業務に関係のある外線電話番号、内線電話番号、郵便の宛名、よく参照する数表の一部、アスキーコード表、また、他のコード表、JIS にあるコード表のコピーなどです。パソコンにデータをまとめることもしますが、独特の使い勝手があります。大学などの研究機関そして一般企業でも、種々の規則や案内をまとめた実用マニュアルを編集しておく必要があります。学校では、身分証明書を含め、規則や年間行事などもまとめて、日記風の手帳を編集して個人ごとに与えることが見られます。対外的には、入試の案内、企業の紹介・入社試験の案内のようなマニュアルを作成しています。実用的な内容がない、マンネリズム的な編集もあって、予算の無駄使いをしている例もあります。